CONTENTS

		Page	
PREFACE		iii	1/A5
I.	PROGRAM SYSTEM OVERVIEW	1	1/A7
	A. Introduction	2	1/A8
	B. Simulation Overview and Nomenclature	3	1/A9
	C. General Use Information	8	1/A14
II.	PROGRAM SEGMENTATION	9	1/B1
III.	DELINEATION OF USER-SUPPLIED MODULES	10	1/B2
	A. Logic Flow	11	1/B3
	B. User-Supplied Module Specifics	12	1/B4
	C. Selected Common-Block Information	17	1/B9
IV.	PROGRAM INPUTS	19	1/B11
	A. Subroutines READ and READIM	19	1/B11
	B. Input Data Stream	20	1/B12
V.	PROGRAM OUTPUTS	20	1/B12
VI.	AUXILIARY PROGRAMS	33	1/C11
	A. REDIM–The Redimension Program	33	1/C11
	B. Support Programs for the DISCOS Flexible-Body Capability	42	1/D6
APPENDIX A–INPUT/OUTPUT SUBROUTINE EXPLANATIONS		A-1	1/F12
APPENDIX B–SUPPORT PROGRAMS FOR THE DISCOS FLEXIBLE-BODY CAPABILITY		B-1	2/A5

Item No. 830-H-5

NAS1.60:1219/v.2

MAY 31 1978

NASA Technical Paper 1219

COMPLETED
ORIGINAL

A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure (DISCOS)

Volume II

Carl S. Bodley, A. Darrell Devers,
A. Colton Park, and Harold P. Frisch

MAY 1978

NASA

464

NAS1.60:1219/V.2

NASA Technical Paper 1219

A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure (DISCOS)

Volume II

Carl S. Bodley, A. Darrell Devers,
and A. Colton Park
Martin Marietta Corporation
Denver, Colorado
Harold P. Frisch
Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1978

All measurement values are expressed in the International System of Units (SI) in accordance with NASA Policy Directive 2220.4, paragraph 4.

PREFACE

This document, prepared by the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, under Contract NAS5-11996, presents the results of a study for the purpose of developing a computer program system for dynamic simulation and stability analysis of passive and actively controlled spacecraft. The study was performed from May 1973 to April 1975 and was administered by the Goddard Space Flight Center, National Aeronautics and Space Administration, Greenbelt, Maryland, under the direction of Joseph P. Young.

Upon delivery, the computer program and associated documentation was checked in detail by Harold P. Frisch. This document incorporates both the original Martin work and the supplementary material prepared at the Goddard Space Flight Center.

The digital computer program, DISCOS (Dynamic Interaction Simulation of Controls and Structure), has been extensively annotated and tested on a range of problems that should have exposed nearly all theoretical errors and programming bugs.

From its inception in 1973, DISCOS has been designed to grow as new needs and more efficient computational techniques developed. This feature makes it impossible to define a final version. To circumvent this problem the official release version will contain only those additions to the delivered program that enhance program documentation and user interface capability and correct programming errors.

Included in this version are more than 10,000 comment cards and a capability to routinely direct the computer to output on the line printer virtually all computation along with explanatory alphanumeric statements. A large percentage of the comment cards are in subroutines DEF1, DEF2, . . ., and DEF5. These subroutines are composed entirely of comment cards and provide the user with an area in the source file for keeping documentation current. In particular, subroutine DEF5 contains a narrative description of the program and its current capabilities.

For the uninitiated reader, it probably would be best to speed-read subroutine DEF5 to obtain a quick overview of the capabilities of the program and the solution techniques applied before reading this document.

The potential user should be aware of the fact that DISCOS is not intended for simple problems. It is primarily for problems which were heretofore intractable. Consequently, the theoretical basis for the program is highly advanced, and the computation algorithms are designed for the efficient processing of the equations associated with large multidegree-of-freedom systems.

As an aid to the user, the paper on which the derivation of the coupled flexible-body equations of motion is based, a paper that outlines the solution method, and comments on results obtained from several DISCOS applications appear as reference material at the end of Volume I. Volume I contains all relevant theoretical work.

Volume 2 describes the DISCOS program and its support programs. The user is encouraged to refer to the comment cards provided in all DISCOS subroutines for additional descriptive information. The comment cards found in the DISCOS subroutines are intended to provide a link between the computer code and the theoretical equations provided in Volume 1.

To effectively interface with the program the user must be able to write the subroutines that will define all non-gyroscope forces and torques. The user is provided with a clean interface. When the load vector associated with the effects of springs, dampers, motors, gas jets, etc. is defined and properly stored in the computer memory, DISCOS will automatically transform it into the appropriate generalized form required by the formulation.

The inclusion of effects such as aerodynamic loading and thermodynamic deformation is more difficult. However, the methodology is analogous to that used for including gravity gradient effects.

The methodology for including loads associated with springs, dampers, motors, gas jets, constraints, etc. is found in Volume II and in the comment cards of the appropriate subroutine referred to in Volume II.

DISCOS is probably the most powerful computational tool to date for the computer simulation of actively controlled coupled multiflexible-body systems. It is not an easy program to understand and effectively apply, but it is not intended for simple problems. The user is expected to have an extensive working knowledge of rigid- and flexible-body dynamics, finite-element techniques, numerical methods, and frequency-domain analysis.

A DIGITAL COMPUTER PROGRAM
FOR THE DYNAMIC INTERACTION SIMULATION OF
CONTROLS AND STRUCTURE (DISCOS)
VOLUME II

Carl S. Bodley, A. Darrell Devers, and A. Colton Park
Martin Marietta Corporation
Denver, Colorado

and

Harold P. Frisch
Goddard Space Flight Center
Greenbelt, Maryland

I. PROGRAM SYSTEM OVERVIEW

This volume is intended to provide sufficient understanding of program system DISCOS (Dynamic Interaction Simulation of Controls and Structure) and its capabilities to permit a user to use the program as a basic tool for analyzing the behavior of a wide range of dynamical problems. Specific emphasis is on a simulation for multiple-interconnected spinning elastic bodies responding under the combined influences of external environments and either active or passive control.

The authors at Martin Marietta Corporation acknowledge the assistance provided by Goddard Space Flight Center personnel. Harold Frisch and James Donohue contributed many valuable technical comments and suggestions throughout the program. In particular, they developed the basic approach to be used for data input, defined exactly what should be included in the transfer function and stability analysis portion of the program, and defined eight of the eleven demonstration problems. They provided the associated data that were used to verify both the nonlinear time response and linear transfer function and the stability analysis portions of the program. Raymond Welch provided the subroutine used to generate root-locus plots. Dr. William Case provided valuable advice on interfacing with NASTRAN output and generated the demonstration problem used to validate the interface subroutine (NASFOR).* During the early program development stage, Dr. James Mason offered significant advice on the need to compute internal forces at the interconnect points. Reginald

---

*The NASFOR subroutine has been superseded by a special NASTRAN DMAP program and associated preprocessor program written by Harold P. Frisch.

Mitchell contributed invaluable advice on requirements for making the program compatible with the GSFC IBM 360/95 computer system. In addition, he supplied the contractor with a 360-compatible plot package, furnished the contractor with a self-authored subroutine for reading NASTRAN output, and was responsible for running all demonstration problems on the 360/95 computer. Finally, the authors acknowledge the encouragement and efforts of Joseph P. Young, Technical Monitor, who made numerous valuable comments and suggestions throughout the study.

A. Introduction

The simulation employs a state-space approach that was developed in detail in Volume I. The state-space formulation provides an attractive basis for simulating nonlinear dynamical problems in a general sense, as well as permitting linearization of the governing equations to provide an additional foundation with which to evaluate frequency-domain and linearized time-domain characteristics.

An attempt has been made to relieve the user of having to communicate with the digital program by means of large amounts of bulk data input. Although many options are available, the program data stream has been organized to require only a minimal amount of basic input data for a particular simulation. The data requirements have been further consolidated in a manner that is quite definitive for the physical system being simulated. In summary, the user can easily relate to the particular elements of the program requirements and, thus, minimize the setup time required for preparing data input for a given problem. In addition, a set of self-checking features has been included for identifying and checking certain compatabilities that are necessary for a proper simulation of a physically realizable system.

In an overall sense, the user can use the digital program to obtain:

- Nonlinear time response
- Interaction constraint forces
- Total system resonance properties
- Frequency domain response and stability information
- Linearized time response

The program outputs consist of printed and plotted* results depicting:

- Dynamic model construction
- Time-domain response
- Frequency-domain characteristics

---

*Plotting uses SC 4020 plot routines that are resident in the IBM-360 system at the Goddard Space Flight Center.

The printed outputs are of a fixed form, and the user controls the plotted information through the input data stream.*

B. Simulation Overview and Nomenclature

This discussion identifies the basic nomenclature used for synthesizing a typical assembly of interconnected bodies. The theoretical development and program annotation make extensive reference to various terminologies that are clarified here. Figure 1 provides a visual display that illustrates many of the items being discussed and will be referred repeatedly to in the ensuing discussions.

The user identifies the overall system "topology" by means of the input integer array, ITOPOL, that contains the necessary information describing which "hinges" interface which bodies. Each body contains a body reference point that is the origin of an orthogonal Cartesian body axis system. This point need not coincide with the body center of mass.

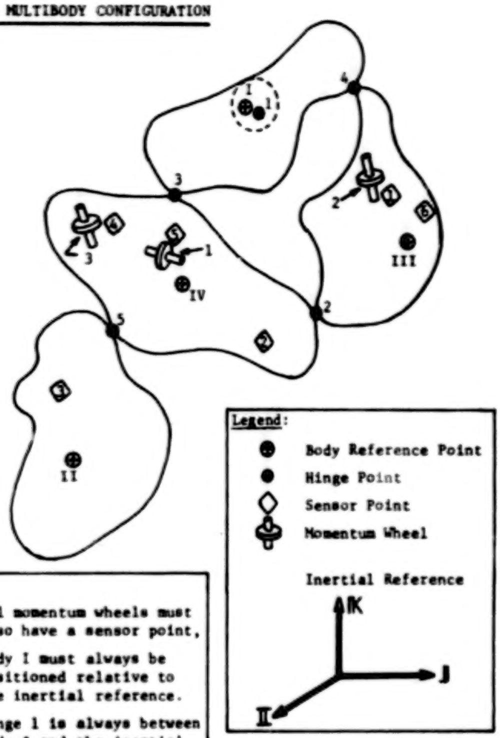
Contiguous "bodies" are interfaced through a "hinge." "Interfaced" is used in lieu of "connected" to emphasize that the common "hinge" point between contiguous bodies may actually permit relative translational motion of the two bodies at the hinge. The user identifies the degree of fixity at the hinge by means of the input integer array, IHDATA. A typical body may contain "sensor" points that identify particular points at which additional information is required for completing the desired simulation. Although a sensor point might sense on position or rate for the control-system inputs, it could also represent a point on a body at which certain other information is desired, such as a momentum-wheel location or a point-of-force/torque application. The integer input array, IFTSMW, identifies the body on which a particular sensor point is located.

A body may also contain "momentum wheels." This special consideration accommodates a disk or rotating mass with a single relative rotational degree of freedom into the simulation without introducing another body. The momentum-wheel capability is more efficient for simulating a single degree-of-freedom rotating mass than for constraining five of six rigid-body degrees of freedom by constraint equations. All momentum wheels must have an associated sensor point. A wheel may be either active or constant speed. An active wheel has a variable spin rate and receives an input torque (usually by some sensor output relationship). A shaft torque is applied to the wheel inducing an angular wheel acceleration. The array, IMO, identifies whether or not the wheel is active, and which axis is the spin axis. The reference axis for the wheel is the same as the sensor-point axis system on which the wheel is located. The array, AMO, identifies the wheel spin rates (initial rate only for the active wheel) and the wheel spin inertia about the spin axis.

The system state vector is arranged in a specific manner within the program, and the user must be very familiar with this arrangement for a number of reasons. First, the user must

---

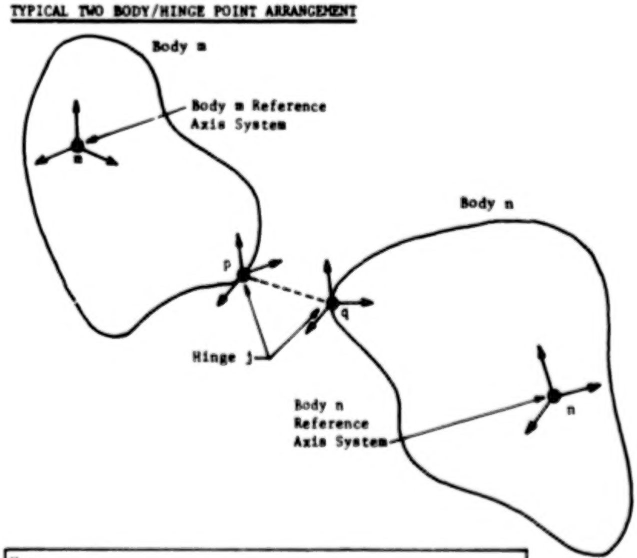
*An option exists for printing most of the significant computations in the primary subroutines if desired.


TYPICAL MULTIBODY CONFIGURATION

I
1
4
3
2
III
IV
II
5
1
2
3
4
5
6
Legend:
Body Reference Point
Hinge Point
Sensor Point
Momentum Wheel
Inertial Reference
K
J
I

Note:

1. All momentum wheels must also have a sensor point,
2. Body I must always be positioned relative to the inertial reference.
3. Hinge 1 is always between Body I and the inertial reference.


TYPICAL TWO BODY/HINGE POINT ARRANGEMENT

Body m
Body m Reference Axis System
m
Body n
p
q
Hinge j
Body n Reference Axis System
n

Note:

1. Motion of the q frame on Body n is measured relative to the p frame on Body m,
2. The p and q frames are positioned with respect to Body m and n reference axis systems,
3. Hinge j is identified via array ITOPOL,
4. Array IHDATA contains additional hinge information.

Figure 1. Simulation nomenclature (Sheet 1 of 4).

PROGRAM SIMULATION INTEGER ARRAYS

ITOPOL Topology, Size = 2 x NH

Hinge No.	1	2	3	4	5	
Body No.	1	3	4	3	2	Body n has frame q
Body No.	0	4	1	1	4	Body m has frame p

Example: Refer to Column 3 (Hinge 3) and note Body 4 interfaces with Body 1 via Hinge 3.

IRGFLX No. of Elastic Modes/Body, Size = 1 x NB

Body No.	1	2	3	4
	5	0	0	2

Example: Body No. 4 has 2 elastic modes, Body No. 2 has no elastic modes.

IPTSMW Sensor Point Locations, Size = 1 x NS

Sensor Pt.	1	2	3	4	5	6
	3	4	2	4	4	3

Example: Sensor Point ⑤ is located on Body 4 (as are Sensor Points ② and ④)

IHDATA Hinge Information, Size = 7 x NH

Hinge No.	1	2	3	4	5
ITYPE	3	1	7	2	11
$\dot{\theta}_1$	0	0	0	1	0
$\dot{\theta}_2$	0	0	0	1	2
$\dot{\theta}_3$	0	0	0	0	1
$\dot{x}$	0	1	1	1	1
$\dot{y}$	0	1	0	1	1
$\dot{z}$	0	1	1	1	0

Note:

1. ITYPE = Euler Type
2. Elements in rows 2-7 identify hinge constraint type.

0: Free/forced
1: Fixed Constraint (Zero relative velocity)
2: Rheonomic Constraint (time dependent)

Additional Remarks:

1. The number of Betas in the state vector equals the sum of "zeros" plus the sum of the "twos" (excluding row 1) in the array.
2. The number of Lambdas (constraints) equals the number of "nonzeros" (excluding row 1) in the array.

IMO Momentum Wheel Information, Size = 3 x NOFMO and
AMO 2 x NOFMO

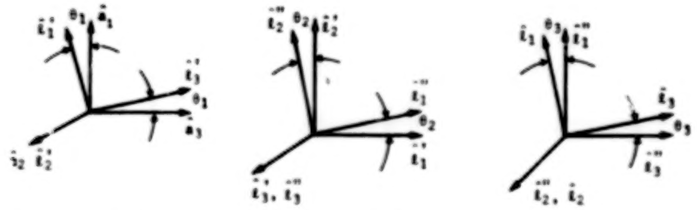
1	2	3	Momentum Wheel No.
5	1	4	Sensor Point No.
2	1	3	Spin Axis (1, 2, or 3)
1	1	0	1 = Active; 0 = Constant Speed
$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	Wheel Rates (Initial)
J_1	J_2	J_3	Wheel Inertia about Spin Axis

Figure 1. Simulation nomenclature (Sheet 2 of 4).

EULER ANGLE PERMUTATION CANDIDATES ∿ ITYPE

	ITYPE =	1	2	3	4	5	6	7	8	9	10	11	12
1st Rotation	Axis of θ_1	1	1	1	1	2	2	2	2	3	3	3	3
2nd Rotation	Axis of θ_2	2	2	3	3	3	3	1	1	1	1	2	2
3rd Rotation	Axis of θ_3	3	1	1	2	1	2	2	3	2	3	3	1

Example: Using ITYPE = 6


$\hat{i}_1'$
θ_1
$\hat{a}_1$
$\hat{i}_3'$
θ_1
$\hat{a}_3$
$\hat{a}_2$ $\hat{i}_2'$
$\hat{i}_2''$
θ_2
$\hat{i}_2'$
$\hat{i}_1''$
θ_2
$\hat{i}_1'$
$\hat{i}_3'$, $\hat{i}_3''$
$\hat{i}_1$
θ_3
$\hat{i}_1''$
$\hat{i}_3$
θ_3
$\hat{i}_3''$
$\hat{i}_2''$, $\hat{i}_2$

θ_1 about axis 2 θ_2 about axis 3 θ_3 about axis 2

$$\begin{Bmatrix}\hat{i}_1\\ \hat{i}_2\\ \hat{i}_3\end{Bmatrix} = \left[\begin{array}{c|c|c}C_3&0&-S_3\\0&1&0\\S_3&0&C_3\end{array}\right]\left[\begin{array}{c|c|c}C_2&S_2&0\\-S_2&C_2&0\\0&0&1\end{array}\right]\left[\begin{array}{c|c|c}C_1&0&-S_1\\0&1&0\\S_1&0&C_1\end{array}\right]\begin{Bmatrix}\hat{a}_1\\ \hat{a}_2\\ \hat{a}_3\end{Bmatrix}$$

also

$$\hat{i}_1 = {}_1R_3\,\hat{i}_3$$

CONSOLIDATION OF KINEMATICAL COEFFICIENTS

(The "b" Coefficients)

$$\begin{array}{c|c|c|c|c}
\text{Hinge No.}\backslash\text{Body No.} & 1 & 2 & 3 & 4\\
\hline
1 & q & & & \\
\hline
2 & & & q & p\\
\hline
3 & p & & & q\\
\hline
4 & p & & q & \\
\hline
5 & & q & & p
\end{array}
\begin{Bmatrix}(U)_1\\ (U)_2\\ \\ (U)_3\\ (U)_4\end{Bmatrix} = \begin{Bmatrix}(\dot{B})_1\\ (\dot{B})_2\\ \\ (\dot{B})_3\\ (\dot{B})_4\end{Bmatrix}$$

Note:

1. Only a single entry is in partition for Hinge 1 and it is a q type partition from the connection kinematical array above,
2. All other row partitions have both a p and a q constituent for each "hinge" between contiguous bodies.

Figure 1. Simulation nomenclature (Sheet 3 of 4).

CONNECTION KINEMATICS ∿ TYPICAL HINGE

$$\begin{Bmatrix}\Delta\dot{\theta}_1\\ \Delta\dot{\theta}_2\\ \Delta\dot{\theta}_3\\ \Delta\dot{x}_1\\ \Delta\dot{x}_2\\ \Delta\dot{x}_3\end{Bmatrix} = \left(\overbrace{\begin{array}{c|c|c} -\tau^{-1}\,{}_{q}R_{p}\,{}_{p}R_{m} & & -\tau^{-1}\,{}_{q}R_{p}\,{}_{p}R_{m}\,\sigma_{p} \\ \hline -{}_{p}R_{m}\,S_{mp} & -{}_{p}R_{m} & -{}_{p}R_{m}\,h_{p} \end{array}}^{\text{a "p" Point}}\middle|\overbrace{\begin{array}{c|c|c} +\tau^{-1}\,{}_{q}R_{n} & & +\tau^{-1}\,{}_{q}R_{n}\,\sigma_{q} \\ \hline {}_{p}R_{q}\,{}_{q}R_{n}\,S_{nq} & {}_{p}R_{q}\,{}_{q}R_{n} & {}_{p}R_{q}\,{}_{q}R_{n}\,h_{q} \end{array}}^{\text{a "q" Point}}\right)\begin{Bmatrix}\{\mathcal{U}_n\}\\ \{\mathcal{U}_m\}\end{Bmatrix}$$

Note:

1. ${}_{i}R_{j}$ is a rotation transformation from the j to the i reference frame.
2. In general, ${}_{i}R_{j}\,{}_{j}R_{k} = {}_{i}R_{k}$ and

τ^{-1} is a transformation relating Euler rates to body axes projections of the angular velocity vector,

σ_i is a modal slope matrix in the i frame,

h_i is a modal deflection matrix in the i frame,

S_{ij} is a skew symmetric matrix whose elements are the components of the vector i to j in the ith frame.

$$\{\mathcal{U}_j\} = \begin{Bmatrix}\omega_x\\ \omega_y\\ \omega_z\\ u\\ v\\ w\\ \dot{\zeta}_1\\ \dot{\zeta}_2\\ \cdot\\ \cdot\\ \cdot\\ \dot{\zeta}_n\end{Bmatrix}_j$$

STATE VECTOR ARRANGEMENT

$$\{y\} = \left\{\begin{array}{c}(U)_1\\ (U)_2\\ (U)_3\\ (U)_4\\ \hdashline (\zeta)_1\\ (\zeta)_2\\ (\zeta)_3\\ (\zeta)_4\\ \hdashline \theta_1\\ \theta_2\\ \theta_3\\ \theta_4\\ \theta_5\\ \theta_6\\ \theta_7\\ \theta_8\\ \theta_9\\ \theta_{10}\\ \theta_{11}\\ \theta_{12}\\ \theta_{13}\\ \theta_{14}\\ \theta_{15}\\ \theta_{16}\\ \theta_{17}\\ \hdashline \delta_1\\ \delta_2\\ \delta_3\end{array}\right\} \;;\; \{\dot{y}\} = \left\{\begin{array}{c}(\dot{U})_1\\ (\dot{U})_2\\ (\dot{U})_3\\ (\dot{U})_4\\ \hdashline (\dot{\zeta})_1\\ (\dot{\zeta})_2\\ (\dot{\zeta})_3\\ (\dot{\zeta})_4\\ \hdashline \dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\\ \dot{\theta}_4\\ \dot{\theta}_5\\ \dot{\theta}_6\\ \dot{\theta}_7\\ \dot{\theta}_8\\ \dot{\theta}_9\\ \dot{\theta}_{10}\\ \dot{\theta}_{11}\\ \dot{\theta}_{12}\\ \dot{\theta}_{13}\\ \dot{\theta}_{14}\\ \dot{\theta}_{15}\\ \dot{\theta}_{16}\\ \dot{\theta}_{17}\\ \hdashline \dot{\delta}_1\\ \dot{\delta}_2\\ \dot{\delta}_3\end{array}\right\} \quad \text{where } \{U\}_j = \left\{\begin{array}{c}\omega_x\\ \omega_y\\ \omega_z\\ \hdashline u\\ v\\ w\\ \hdashline \dot{\zeta}_1\\ \dot{\zeta}_2\\ \cdot\\ \cdot\\ \cdot\\ \dot{\zeta}_{UE}\\ \hdashline \dot{\theta}_1\\ \dot{\theta}_2\\ \cdot\\ \cdot\\ \cdot\\ \dot{\theta}_M\end{array}\right\}_j$$

← For this example, prescribed as function of time, i.e., a rheonomic constraint as noted in INDATA

CONSTRAINT FORCE/TORQUE VECTOR

$$\{\lambda\} = \left\{\begin{array}{c}\lambda_1\\ \lambda_2\\ \lambda_3\\ \hdashline \lambda_4\\ \lambda_5\\ \hdashline \lambda_6\\ \lambda_7\\ \hdashline \lambda_8\\ \lambda_9\\ \lambda_{10}\\ \hdashline \lambda_{11}\\ \lambda_{12}\\ \hdashline \lambda_{13}\\ \lambda_{14}\end{array}\right\} \begin{array}{l}(F_x)_2\\ (F_y)_2\\ (F_z)_2\\ (F_x)_3\\ (F_z)_3\\ (T_1)_4\\ (T_2)_4\\ (F_x)_4\\ (F_y)_4\\ (F_z)_4\\ (T_2)_5\\ (T_3)_5\\ (F_x)_5\\ (F_y)_5\end{array}$$

$(F_x)_i$ refers to Hinge i

Figure 1. Simulation nomenclature (Sheet 4 of 4).

know where certain variables are located so that he can couple the control law into the simulation, and, secondly, the user must know the order of the state variables in order to interpret results. Figure 1 presents the state variable order consistent with the illustrative problem and other related information. The state variables shown represent a typical arrangement in that all of the various types of variables resulting from the multiple options available within the simulation are present. The order of the constraints, λ, is also noted. Note that, although the user introduces the control variables into the state vector, these variables, δ, will always appear after the betas, β. Furthermore, the user may also introduce auxiliary variables (plant sensor signals and control-system outputs) for use in the linearized studies. These auxiliary variables should be placed (by the user) after the control variables, δ, and in the order: plant sensor signals, X_{ss}, followed by control-system outputs, B.

C. General Use Information

A number of guidelines must be followed in setting up a particular simulation. Some were detailed previously and are concisely summarized here:

- There must be at least two bodies; a single-body problem is simulated by including a dummy body that is not connected to the body to be analyzed.
- Body 1 is always positioned relative to the inertial reference.
- Bodies are numbered from 1 to NB in an arbitrary order.
- Each body (except body 1) must have at least *one* hinge; body 1 must have at least two hinges.
- Hinges are numbered from 1 to NH in an arbitrary order, but hinge 1 is, by definition, the hinge on body 1 between body 1 and the inertial origin; hence, hinge 1 can only appear on body 1.
- There must be at least one sensor point for a given simulation.
- Sensor points are numbered from 1 to NS in an arbitrary manner.
- A typical flexible body requires mass and modal data that reflects a coordinate system that is consistent with the body axis reference system for that body (e.g., a modal coupling approach that establishes modal properties for a given body would have to use the same reference body axis system).
- For frequency-domain studies, only as many control output variables can be identified for introducing into the state equations as there are control-system variables to begin with. Similarly, no more sensor-signal variables can be identified than plant variables that appear in the original independent state equations.
- The user must ensure that the user-supplied package has dimensions consistent with NHMAX for the following arrays:

  SK(NSK, NHMAX), DK(NDK, NHMAX), and HNGT(NHT, NHMAX) where

NHMAX = dimensioned maximum number of hinges

NSK = 3 or 6, depending on the nature of hinge freedom

NDK = 3 or 6, depending on the nature of hinge freedom

NHT = 3 or 6, depending on the nature of hinge freedom

and, if rotation only, then NSK = NDK = NHT = 3; if rotation and translation, then NSK = NDK = NHT = 6.

- The intertial properties of all the momentum wheels in a particular body must be included in that body's inertia description (whether rigid or flexible) because inertial coupling is used.

II. PROGRAM SEGMENTATION

The digital code has been segmented into an executive overlay that governs the succeeding program flow and four supporting primary overlays, each with a separate and dedicated purpose. The basic program flow is depicted in figure 2.

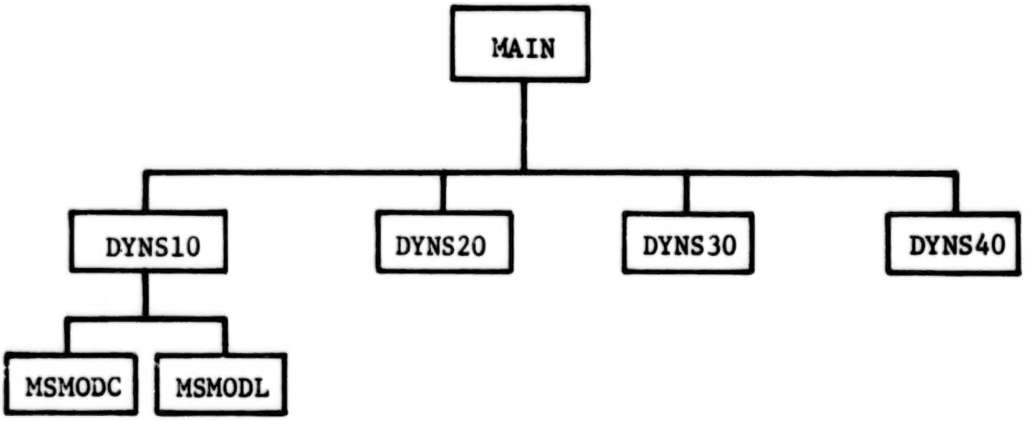


Figure 2. DISCOS program segmentation.

Table 1 summarizes the intended purpose of the fundamental components in the program structure.

The executive overlay (MAIN) initiates the simulation by reading job-identification information and then passes control to the first primary overlay (DYNS10), which represents the basic data input segment. This overlay may be viewed as the program segment that builds the model from the input data. A series of topology checks are made as the data are loaded within this overlay to better ensure proper modeling of the physical system. This overlay utilizes two additional secondary overlays for processing certain types of inertial and modal data.

Table 1
DISCOS Overlay Description

Segment Name	Purpose
Primary Overlays	
MAIN	Executive program control
DYNS10	Data input
DYNS20	Simulation of problem, linearization of state equations, and nonlinear time response
DYNS30	Plot results from nonlinear or linearized time response
DYNS40	Frequency-domain analysis, linearized time response, and frequency-domain displays (Bode, Nichols, Nyquist, Root Locus)
Secondary Overlays (called from DYNS10)	
MSMODL	Flexible-body data inputs for lumped mass representation
MSMODC	Flexible-body data inputs for consistent mass representation

After overlay DYNS10 has structured the basic data for simulation, control is returned to the executive overlay that, in turn, passes control to the second overlay, DYNS20. This overlay performs the actual problem mechanization and develops the nonlinear formulation that is the foundation for the entire dynamic simulation program.

During a given simulation, the executive overlay always calls both the first and the second primary overlays, DYNS10 and DYNS20, but, depending on certain input control parameters, may or may not call the time-history plot overlay, DYNS30, or the linearized system-analysis overlay, DYNS40.

Simulation of a particular problem has its basis within the algorithms contained in the program subroutine, YDOT, which establishes the canonical first-order differential equations that govern the dynamical motion. In turn, this routine addresses another subprogram, TORQUE, which, in turn, activates the user-supplied modules that relate to the particular simulation being considered. The following section describes these modules in further detail.

III. DELINEATION OF USER-SUPPLIED MODULES

The program has been written under the assumption that certain user-supplied modules are available for completing a given problem. In this manner, the user has considerable latitude concerning how certain particulars related to a given simulation should be handled. Control-law specification, external torque inputs, and identification of plant sensor signals and control-system outputs are examples of items handled by the user. With this concept in mind, several subprograms have been placed under user control, but with certain restrictions and guidelines to which the user must adhere. Later comments will identify certain requirements associated with these user-supplied modules.

A. Logic Flow

It is worthwhile to consider a flow-chart segment of the program (figure 3) and its chronology within the solution process. The order in which the user-supplied modules are called is indicated by integers 1 through 7 for subroutines and 8 and 9 for functions.

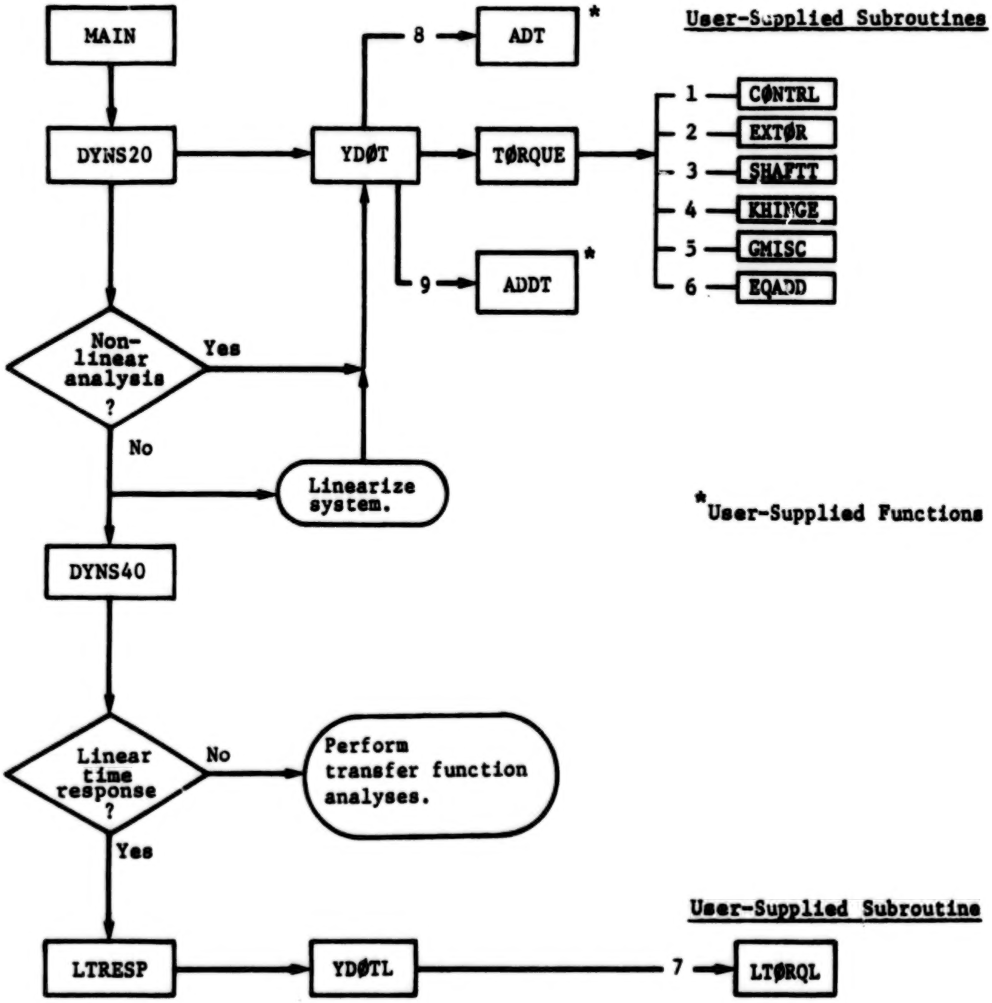

MAIN
DYNS20
YDØT
ADT
TØRQUE
ADDT
User-Supplied Subroutines
1 CØNTRL
2 EXTØR
3 SHAFTT
4 KHINGE
5 GMISC
6 EQADD
8
9
*
Non-linear analysis ?
Yes
No
Linearize system.
*User-Supplied Functions
DYNS40
Linear time response ?
No
Perform transfer function analyses.
Yes
User-Supplied Subroutine
LTRESP
YDØTL
7
LTØRQL

Figure 3. Chronology of addressing user-pack.

B. User-Supplied Module Specifics

Each user-supplied subprogram has specific intended purposes and has been coded to fulfill these goals. The user can extend the scope of any of these modules with his own code, but these routines must perform certain tasks. Because the potential user should be familiar with many details of the user-supplied package, the following paragraphs are devoted to each of the user-supplied modules. Because reference will be made to some of the programming logic contained in the DISCOS subroutine TORQUE, this logic has been put into flow-chart form as figure 4. Additional comments and rules to be followed in coding the user modules can be found in the comment cards included in the program listing of the default form of each of the routines and in subroutine DEF5.

1. CONTRL

CONTRL is the first of the user-supplied subroutines and is always called by DYNS20. The primary purpose of CONTRL is to establish the time derivatives of the control-system variables. These variables may be required by some of the other user routines that are activated after CONTRL has been addressed. The routine must also establish the number of plant sensor signals, NXSS, and the number of control-system outputs, NBTQ, that are transmitted through the common block, /LDSIZE/,* to the remainder of the program. For transfer-function studies, the user must also identify whether or not transfer-function polynomials are to be used. This is accomplished in a data statement (in CONTRL) with the variable, NPLY, which is the number of polynomial ratio pairs (numerator and denominator) to be used. For NPLY $\neq 0$, the first call to CONTRL will read in the polynomial coefficients.

Subroutine CONTRL contains a good deal of information pertaining to the simulation by virtue of its common blocks. Section III-C identifies the constituents of the common blocks contained in this and other modules. The program user can establish additional common blocks to transfer information between the separate user-pak modules.†

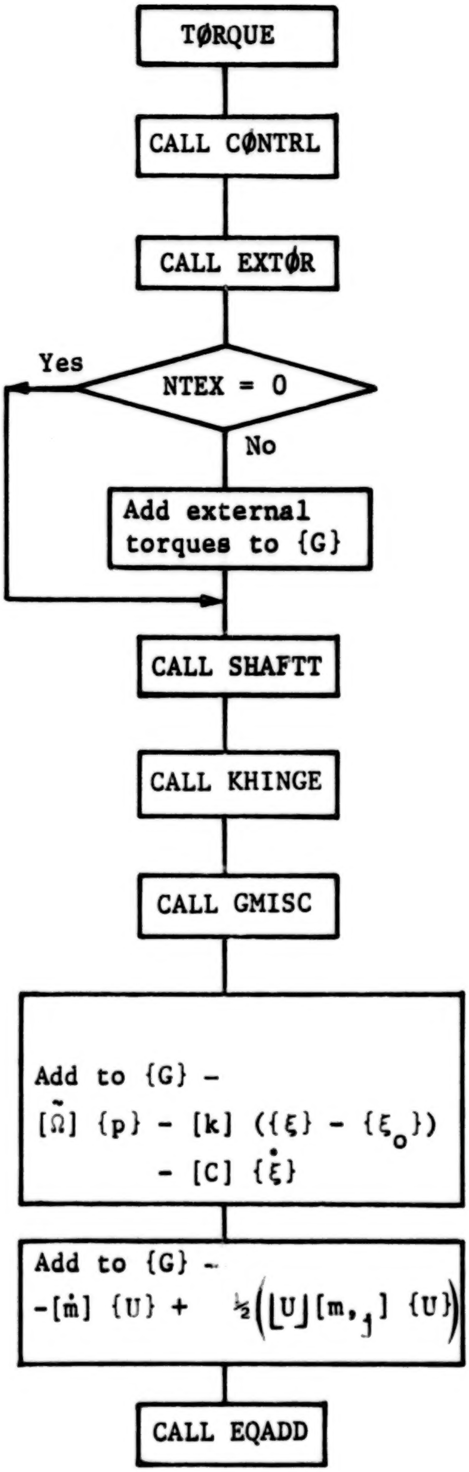
2. EXTOR

Subroutine EXTOR establishes the external system torques. Typically, this module can accommodate items such as reaction-control system (RCS) forces and torques, aerodynamics, and/or solar wind. The user can also extend this routine to include other state-dependent torques. In summary, EXTOR can be used as a "catchall" for including any additional forces and torques that affect the system. A single call to EXTOR from subroutine TORQUE establishes an integer array, ISNP, whose elements identify which sensor points are to be used for force/torque inputs. A vector containing torque and force components (ordered: T_x, T_y, T_z, F_x, F_y, F_z) is then established for each of the force/torque sensor points and is

---

*Internal logic loops require that NXSS be set greater than zero.

†See subroutine DEF1 and DEF5.


TØRQUE
CALL CØNTRL
CALL EXTØR
Yes
NTEX = 0
No
Add external torques to {G}
CALL SHAFTT
CALL KHINGE
CALL GMISC
Add to {G} -
$[\tilde{\Omega}]\{p\} - [k](\{\xi\} - \{\xi_o\}) - [C]\{\dot{\xi}\}$
Add to {G} -
$-[\dot{m}]\{U\} + \frac{1}{2}\left(\lfloor U \rfloor [m,_j]\{U\}\right)$
CALL EQADD

Creates external torques/forces

Note: {G} is force/torque vector ∿ RHS of equation of motion

Adds momentum wheel shaft torques to {G}

Adds hinge spring torques to {G} and sums spring energy to potential energy

Gets $\Delta\xi$ due to thermal environment

Now have RHS of equations of motion

Creates additional equations for similarity transformation (used in linearization and stability package)

Figure 4. Subroutine TORQUE flow diagram.

placed as a column into the array, TEX. The vector of discrete forces and torques (referred to the local sensor-axis system) is returned to subroutine TORQUE from EXTOR. These forces and torques are then transformed and added to the total system external force/torque array $\{G\}$. The user can bypass EXTOR-related calculations by setting the variable, NTEX, equal to zero.

3. SHAFTT

Subroutine SHAFTT establishes the shaft torque for each of the nonconstant-speed momentum wheels. Zeros are inserted for the torque contributions to the external torque vector for a constant-speed wheel.

4. KHINGE

Subroutine KHINGE sets up hinge-spring and dashpot torques/forces and accounts for potential energy contributions caused by hinge-spring deflections. The user must identify where spring rates and dashpot constants should be found. This can be easily handled by a user-specified equivalence statement within subroutine KHINGE to locate the leading stiffness and damping elements within the data block identified as CNTDTA. Note that subroutine KHINGE (see subroutine listing) contains statements of the following form:

```
DIMENSION SK(3, NHMAX), DK(3, NHMAX), HNGT(3, NHMAX)
      .
      .
      .
DO 10 I=1,3
      .
      .
DO 15 I=1,3
      .
      .
DO 20 I=1,3
      .
      .
```

where integer 3 reflects the fact that consideration has been restricted to admitting only rotational springs at each hinge. If the user wants to also include springs or dashpots in relative translation at the hinge points, the 3 in the foregoing statements must be changed to a 6, and appropriate spring rates and/or dashpots included within the data input array, CNTDTA. Further, the equivalence statement that locates the first spring rate, SK(1), and dashpot constant, DK(1), reflect an order that is consistent with the hinge order (i.e., the first three elements in CNTDTA beginning with the location corresponding to the leading element of SK represents in order K_{θ_1}, K_{θ_2}, and K_{θ_3} for the first hinge). A similar relationship exists for the array, DK. For example, in KHINGE, note that:

EQUIVALENCE(CNTDTA(K),SK(1)),(CNTDTA(L),DK(1))

and the array, CNTDTA, would be:

$$\text{CNTDTA} \quad = \quad \ldots \underbrace{K_{\theta_1}\ K_{\theta_2}\ K_{\theta_3}}_{\substack{\text{Hinge 1}\\ \text{(springs)}}} \ldots \underbrace{K_{\theta_1}\ K_{\theta_2}\ K_{\theta_3}}_{\substack{\text{Hinge NH}\\ \text{(springs)}}} \ldots$$

element (K)

element (L)

$$\ldots \underbrace{C_{\dot\theta_1}\ C_{\dot\theta_2}\ C_{\dot\theta_3}}_{\substack{\text{Hinge 1}\\ \text{(dashpots)}}} \ldots \underbrace{C_{\dot\theta_1}\ C_{\dot\theta_2}\ C_{\dot\theta_3}}_{\substack{\text{Hinge NH}\\ \text{(dashpots)}}} \ldots$$

where the order of $\theta_{1,2,3}$ is consistent with the Euler rotation type for the hinge "q" triad, specified by user in array IHDATA.

The remainder of KHINGE is concerned with the proper placement of the spring/dashpot forces and torques onto the composite NB bodies (generalization of forces and torques) and should remain unchanged.

The user can modify the referenced torques and forces immediately after the (DO 10 L=1, NH) loop if he desires, but must be careful to ensure that the proper force or torque is correctly applied to accomplish the desired result.

5. GMISC

Subroutine GMISC is reserved for implementing torque/force contributions from thermal gradient effects. The entire state vector, along with component position and attitude information, is available by means of transfer through labeled common arrays. Paragraph III.C provides more detail on the information contained in these common blocks.

6. EQADD

Subroutine EQADD establishes additional equations for use in the linearized time-domain analyses. It must identify the number of additional equations introduced by the variable, NAUX (number of auxiliary equations): These equations relate plant sensor signals, X^i_{ss}, and control-system output forces/torques, B^i, to the system state in the specified order.

The additional variables will be automatically placed in the state vector and become an integral part of system transfer-function evaluations. (See comment cards for subroutines EQADD and DEF5.)

7. LTORQL

Subroutine LTORQL establishes the $b_{ik} U^k$ portion of the right-hand side of

$$\dot{z}^i = A_{ij} z^j + b_{ik} U^k$$

which is used for the linearized time response. This corresponds to the external excitations for the transformed variables, z^i, leading to evaluation of the perturbation response.

8. ADT (Function)

Function ADT is used in conjunction with subfunction ADDT to implement prescribed kinematical motion in the hinge coordinates. With reference to figure 1, the array, IHDATA(I,J),I>1, may have 2 as an entry, indicating that the J^{th} hinge has velocity and acceleration prescribed in that coordinate. The argument of this subfunction is: ADT(IC,T) with IC=6*(J-1)+(I-1) corresponding to IHDATA(I,J)=2. The integer, IC, and the time, T, are passed into ADT via argument by the calling subroutine so that the velocity, $\dot{\alpha}$, may be established for the proper hinge coordinate as a function of time.

For a given rheonomic constraint, note that both $\dot{\alpha}$ and $\ddot{\alpha}$ must be set by subfunction. It is conceivable that the user knows $\ddot{\alpha}$ as the exact mathematical time derivative of $\dot{\alpha}$. It would seem that the natural thing to do is to create ADT and ADDT subfunctions to return consistent $\dot{\alpha}$ and $\ddot{\alpha}$, respectively. This is *not* the best thing to do, however, because of numerical integration characteristics. The numerical integration of $\{\dot{U}\}$ reflects the use of $\ddot{\alpha}$. The resulting $\{U\}$ reflects a numerically integrated $\dot{\alpha}$ that cannot be consistent with a value obtained by any method other than numerical integration. The consequences of this are seen as slight errors in motion response and a large spurious change in system momenta.

The best way to effect rheonomic constraints is to use values of $\dot{\alpha}$ obtained from numerically integrating $\ddot{\alpha}$. This can be easily done by using additional differential equations that are accommodated in the state vector as additional "control variables" or $\{\delta\}$. Thus, when all actual control variable rates are established (in subroutine CONTRL), the user need only code additional expressions to set $\dot{\delta}$ (additional) = $\ddot{\alpha}$ (desired). The statements within subfunction ADT merely return ADT=Y(K); the state vector, Y, is available in labeled common /VECTOR/, and K corresponds to the location in Y where the $\delta = \dot{\alpha}$ control variable resides. Of course, IC must be tested so that the appropriate $\dot{\alpha} = \delta$ is returned.

9. ADDT (Function)

Function ADDT is discussed with regard to its relationship to ADT (paragraph III.B.8). This function has arguments, ADDT(IC,T), identical to those of ADT, and returns values of α for appropriate IC and T consistent with the $\dot{\alpha}$ returned by ADT. Note that, in figure 3, the chronology is such that subroutine CONTRL is addressed before function ADDT so that CONTRL can establish a value of $\dot{\delta}$ (additional) = $\ddot{\alpha}$ (desired) to put in the state-vector time derivative, YDT (also available in labeled common/VECTOR/). Now, for the appropriate time, T, and IC, it is necessary only to set ADDT=YDT(K), where K again corresponds to the location in Y at which the $\delta = \dot{\alpha}$ auxiliary control variable resides.

C. Selected Common-Block Information

To compute specific variables required for the user-supplied modules, the program user will often require certain information that is calculated and stored within the program. Such information about the simulation is stored in multidimensional array form within labeled common blocks. These data provide a good supplement to the state variable content that has been discussed previously in that the user can extract both total and relative positions and rates for any component of the simulated dynamical system if he has a firm understanding of where certain data reside within the program. The following subsections discuss selected common-block arrays to familiarize the potential user with their content.*

1. IBHBSRDI

Common block /BHBSRD/ contains three separate groups of information that the user may require. This information is concisely summarized in double- and/or triple-subscripted arrays as

BS(6,6+NMDBOD,NSPMAX)

ROL(3,3,NBMAX)

DOL(3,NBMAX)

where the following items are noted:

NMDBOD = maximum-dimensioned number of modes per body

NBMAX = maximum-dimensioned number of bodies

NSPMAX = maximum-dimensioned number of sensor points

The array, BS(i,j,k), contains the kinematical coefficients for all of the "sensor" points. The rows (subscript i=1,2...6) of the array refer to (in order: ω_x, ω_y, ω_z, u, v, w) the components of absolute angular and translational velocity (sensor referenced) at sensor point k. The columns of the array (subscript j) refer to the j = 1,2,. . .6 + number of elastic modes on body containing sensor point k. Thus, in general, the following expression can be used to determine the projection (the i^{th} velocity component) onto the triad located at sensor point k:

---

*Also see subroutines DEF1 and DEF5.

$$\mathrm{Vel}_i = \mathrm{BS}\,(i, j_1, \ldots, j_L, k) \cdot \tilde{U}^j$$

The array, ROL(i,j,k), contains the rotation transformations relating the body axis systems to the inertial reference. The elements of the array are the direction cosines between the body axes, e_k, and the fixed inertial system, e_0. Subscript k denotes the body number.

The array, DOL(i,k), contains the three vector components, X, Y, Z), from the inertial reference to the body axis system, e_k, for each body.

2. /SPECIF/

Common block /SPECIF/ contains information that the user may require. These arrays are:

BETAH(6,NHMAX)

BETAHD(6,NHMAX)

RS(3,3,2*(NSPMAX))

DS(3,2*(NSPMAX))

where the following items are noted:

NHMAX = maximum-dimensioned number of hinges

NSPMAX = maximum-dimensioned number of sensor points

The arrays, BETAH(i,j) and BETAHD(i,j), contain the hinge BETA's and rates for hinge j, respectively. The order (i subscript) is given as

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\Delta}_1 \\ \dot{\Delta}_2 \\ \dot{\Delta}_3 \end{bmatrix}$$

where $\dot{\theta}_i$ is the i^{th} Euler angle rate consistent with ITYPE for hinge j, and $\dot{\Delta}_i$ is the i^{th} velocity component of point q relative to point p in the p frame for hinge j.

The array, RS(i,j,k), contains the rotation transformations (direction cosines) between the sensor-point axis system and the body axis system (body on which sensor is located). Two sets of transformations are identified for a given sensor point. The first represents misalignment of the two triads without elastic deformation, and the second includes the elastic deformation. The ordering (subscript k) proceeds as follows: The ℓ^{th} sensor rotation (without elastic deformation) is located at $k = 2*\ell-1$. The total rotation transformation for the ℓ^{th} sensor is located at $k = 2*\ell$. For a rigid body, these two transformations are identical.

The array, DS(i,k), contains the three components of the vector from the body axis system to the body sensor points (in the body axis system). Here again, there are two sets of vectors (rigid body and rigid body + elastic) for each sensor point. The first is for rigid body and the second includes the elastic deformation. The addressing algorithm is the same as that for RS(i,j,k).

IV. PROGRAM INPUTS

The dynamic simulation program utilizes some basic data input subroutines in an attempt to standardize a large amount of the bulk data input. Additional formatted inputs have been used when it is more meaningful (and more efficient) to do so. As stated in the following section, there is a large amount of data input via subroutines READ and READIM. Therefore, it is useful to familiarize the reader with these two routines before describing overall program data input requirements.

A. Subroutines READ and READIM

Subprograms READ and READIM are structured to read matrix arrays in floating-point (real) notation (subroutine READ) or fixed-point (integer) notation (subroutine READIM). A thorough discussion of the routines and their supporting subroutines is contained in Appendix A. The following discussion gives a cursory overview of their use.

The routines are activated by a FORTRAN call of either of the following forms:

```
CALL READ (A, NR, NC, KR, KC)

CALL READIM (IA, NR, NC, KR, KC)
```

where the arguments in the call statement are:

A, (IA) = floating (fixed) matrix array of size NR by NC

NR = number of rows in array

NC = number of columns in array

KR = row dimension of array in calling program

KC = column dimension of array in calling program

A call to either of these input routines requires that the data be in the following formats:

- Subroutine READ

  First card–Matrix name, NR, NC with format (A6, I4, I5)

  Middle cards–Data with format (2I5, 4D17.8)

  First I5 is row number
  Second I5 is column number of leading D17.8 field
  Next 4D17.8 are elements of the array

  Last card–Ten zeros in columns 1 through 10

- Subroutine READIM

  First card–Matrix name, NR, NC with format (A6, I4, I5)

  Middle cards–Data with format (2I5, 14I5)

  First I5 is row number
  Second I5 is column number of leading I5 field
  Next 14I5 are elements of the array

  Last card–Ten zeros in columns 1 through 10

B. Input Data Stream

This section presents the program system input data stream, together with the data input control logic. The approach herein is to first introduce an overview of the data inputs and program control logic in the form of a flow diagram (figure 5) and to then identify the details in much the same way as the FORTRAN code accepts the data inputs. This method of presentation has been chosen because it most closely relates to the actual processing of the user inputs for a given simulation. In addition, the user can follow the program control or switching logic to determine just what data are required for completing a particular simulation.

To keep this documentation current with the program, subroutine DEF3 will contain a detailed description of the input data. As new capability and data input requirements are added to DISCOS, documentation will be kept current by the insertion of suitable comment cards in subroutine DEF3.

V. PROGRAM OUTPUTS

This section discusses the various program output information and correlates the output data with both the input data and the problem simulation. This information is presented in

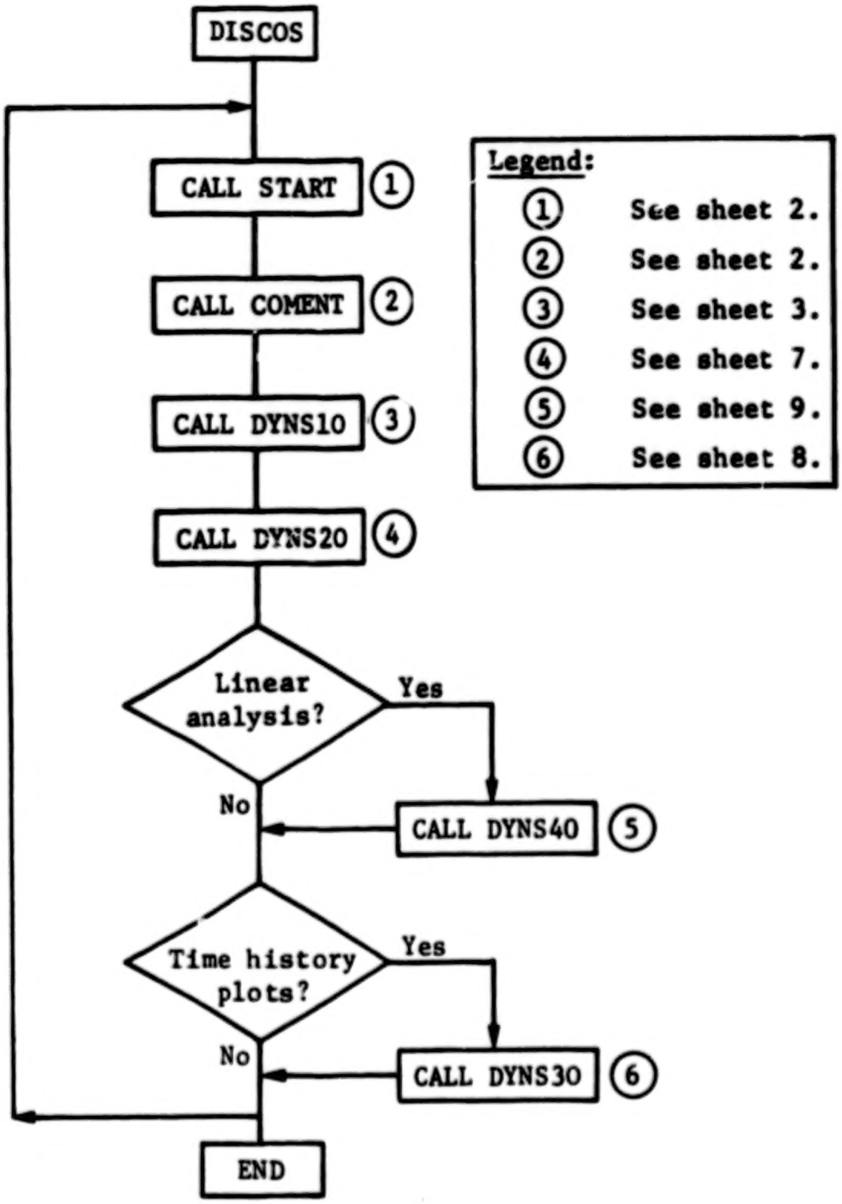

DISCOS
CALL START ①
CALL COMENT ②
CALL DYNS10 ③
CALL DYNS20 ④
Linear analysis?
Yes
No
CALL DYNS40 ⑤
Time history plots?
Yes
No
CALL DYNS30 ⑥
END
Legend:
① See sheet 2.
② See sheet 2.
③ See sheet 3.
④ See sheet 7.
⑤ See sheet 9.
⑥ See sheet 8.

Figure 5. Program system DISCOS data stream flow (Sheet 1 of 12).

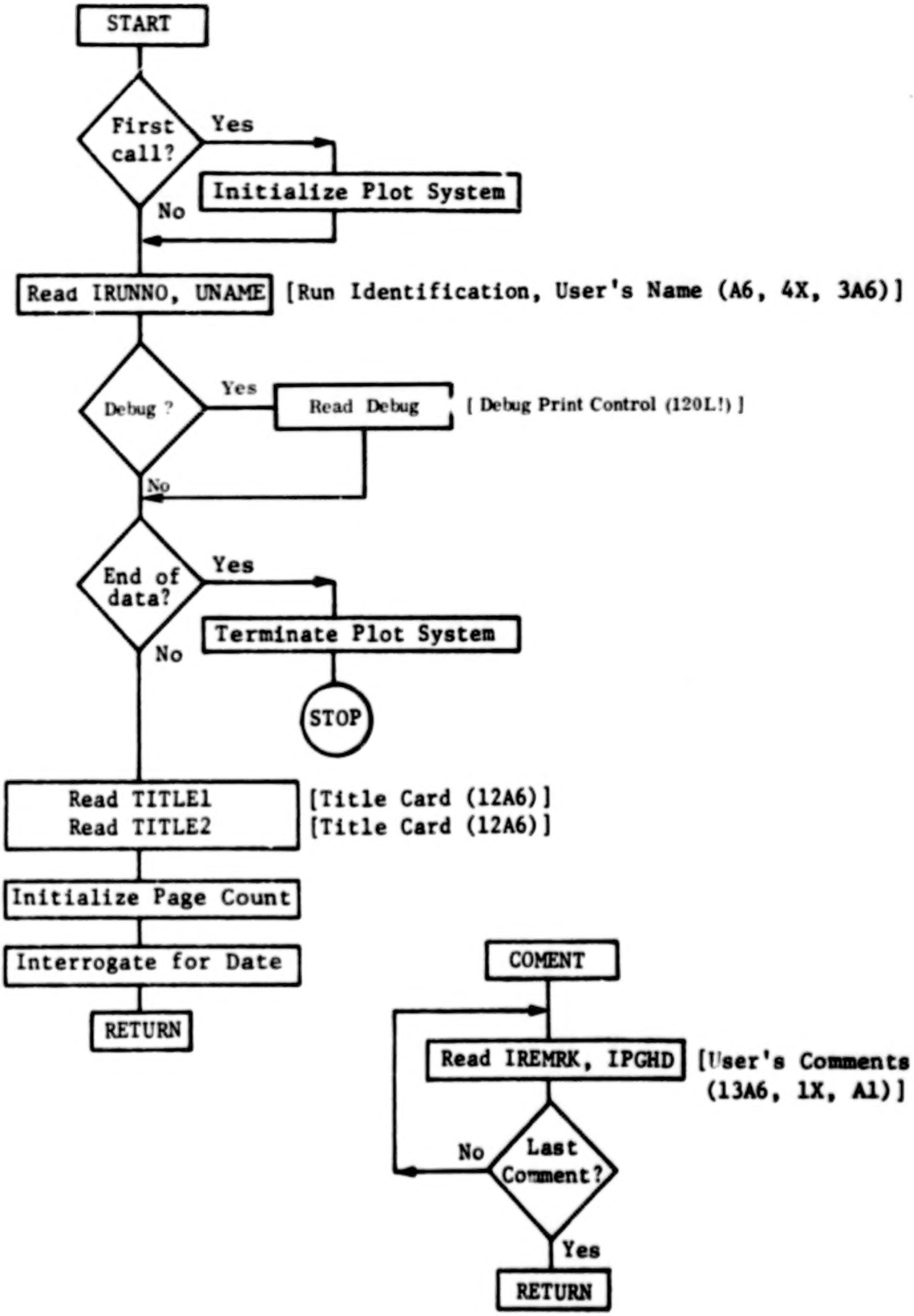

START
First call?
Yes
Initialize Plot System
No
Read IRUNNO, UNAME
[Run Identification, User's Name (A6, 4X, 3A6)]
Debug ?
Yes
Read Debug
[Debug Print Control (120L1)]
No
End of data?
Yes
Terminate Plot System
STOP
No
Read TITLE1
Read TITLE2
[Title Card (12A6)]
[Title Card (12A6)]
Initialize Page Count
Interrogate for Date
RETURN
COMENT
Read IREMRK, IPGHD
[User's Comments (13A6, 1X, A1)]
Last Comment?
No
Yes
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 2 of 12).

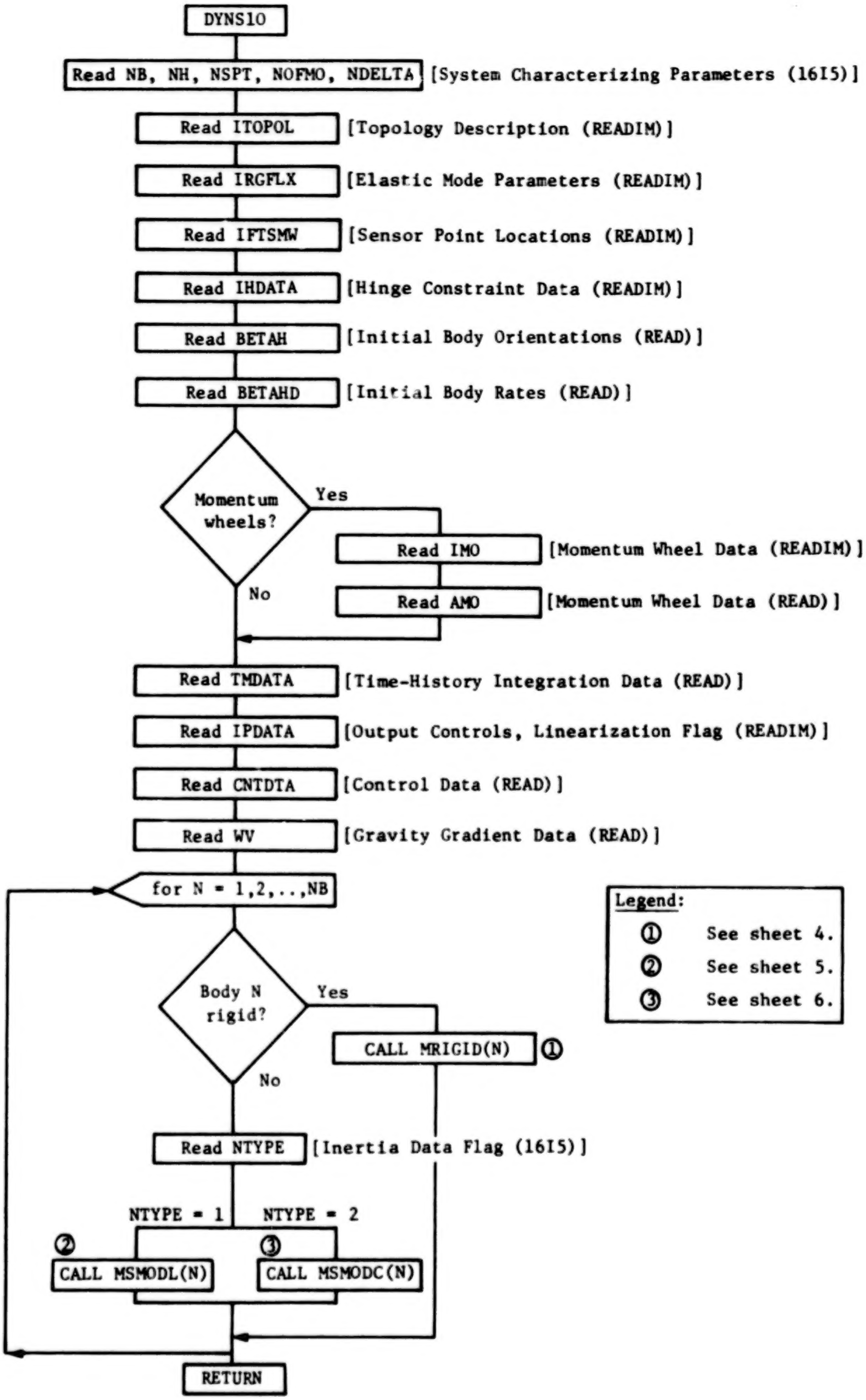

DYNS10
Read NB, NH, NSPT, NOFMO, NDELTA
[System Characterizing Parameters (16I5)]
Read ITOPOL
[Topology Description (READIM)]
Read IRGFLX
[Elastic Mode Parameters (READIM)]
Read IFTSMW
[Sensor Point Locations (READIM)]
Read IHDATA
[Hinge Constraint Data (READIM)]
Read BETAH
[Initial Body Orientations (READ)]
Read BETAHD
[Initial Body Rates (READ)]
Momentum wheels?
Yes
Read IMO
[Momentum Wheel Data (READIM)]
No
Read AMO
[Momentum Wheel Data (READ)]
Read TMDATA
[Time-History Integration Data (READ)]
Read IPDATA
[Output Controls, Linearization Flag (READIM)]
Read CNTDTA
[Control Data (READ)]
Read WV
[Gravity Gradient Data (READ)]
for N = 1,2,..,NB
Legend:
① See sheet 4.
② See sheet 5.
③ See sheet 6.
Body N rigid?
Yes
CALL MRIGID(N) ①
No
Read NTYPE
[Inertia Data Flag (16I5)]
NTYPE = 1
NTYPE = 2
②
③
CALL MSMODL(N)
CALL MSMODC(N)
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 3 of 12).

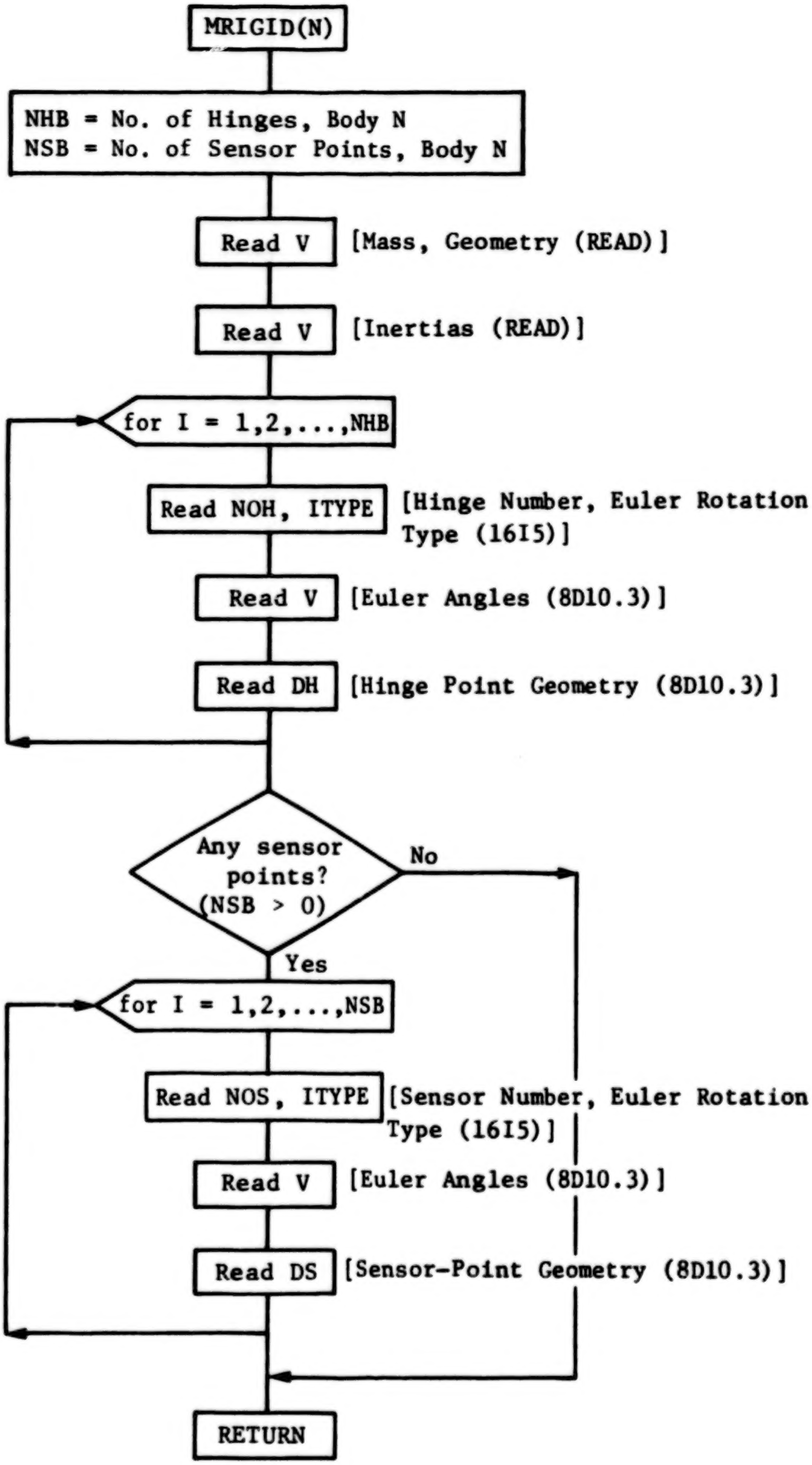

MRIGID(N)
NHB = No. of Hinges, Body N
NSB = No. of Sensor Points, Body N
Read V
[Mass, Geometry (READ)]
Read V
[Inertias (READ)]
for I = 1,2,...,NHB
Read NOH, ITYPE
[Hinge Number, Euler Rotation Type (16I5)]
Read V
[Euler Angles (8D10.3)]
Read DH
[Hinge Point Geometry (8D10.3)]
Any sensor points? (NSB > 0)
No
Yes
for I = 1,2,...,NSB
Read NOS, ITYPE
[Sensor Number, Euler Rotation Type (16I5)]
Read V
[Euler Angles (8D10.3)]
Read DS
[Sensor-Point Geometry (8D10.3)]
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 4 of 12).

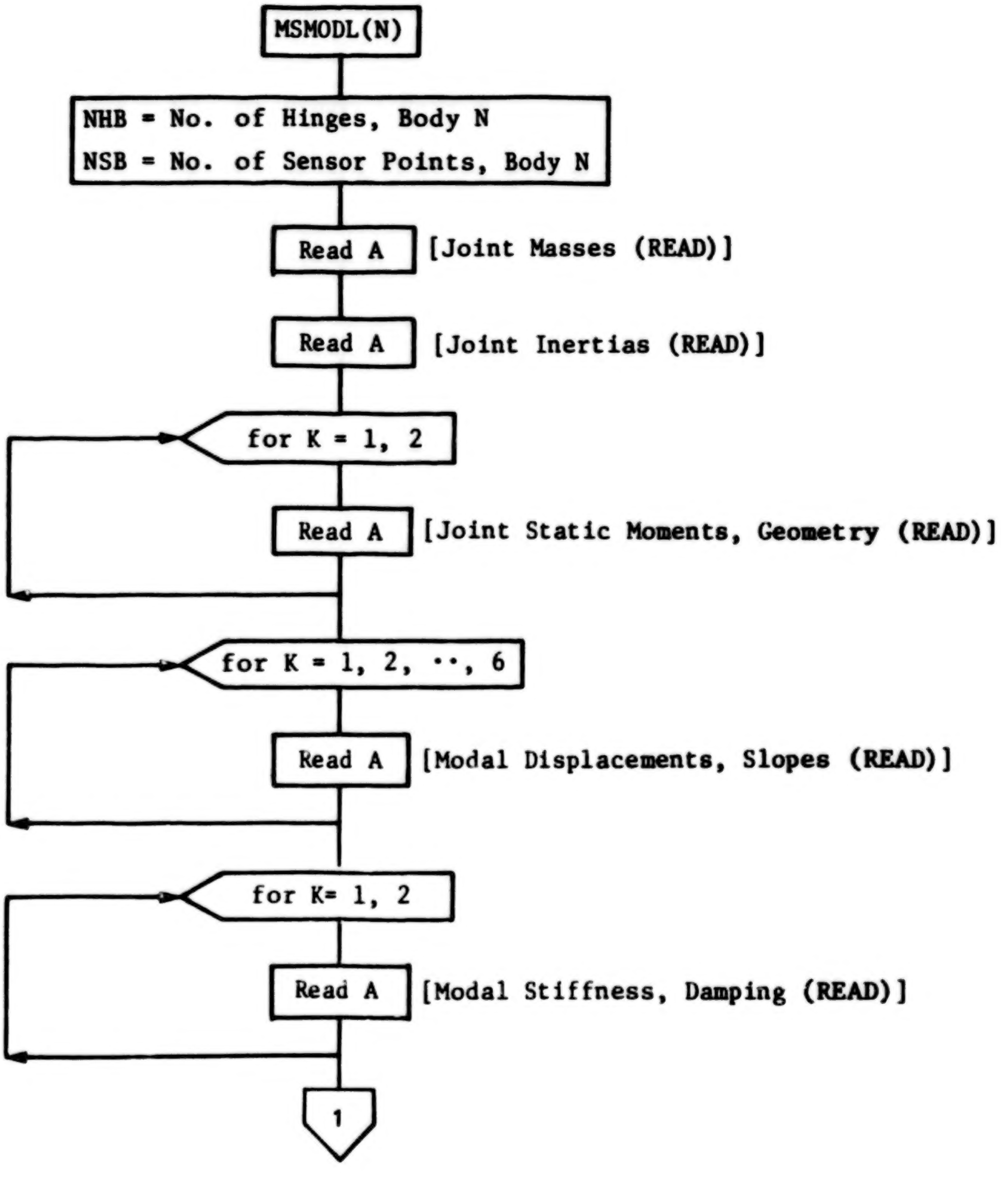

MSMODL(N)
NHB = No. of Hinges, Body N
NSB = No. of Sensor Points, Body N
Read A
[Joint Masses (READ)]
Read A
[Joint Inertias (READ)]
for K = 1, 2
Read A
[Joint Static Moments, Geometry (READ)]
for K = 1, 2, ··, 6
Read A
[Modal Displacements, Slopes (READ)]
for K= 1, 2
Read A
[Modal Stiffness, Damping (READ)]
1

Figure 5. Program system DISCOS data stream flow (Sheet 5 of 12).

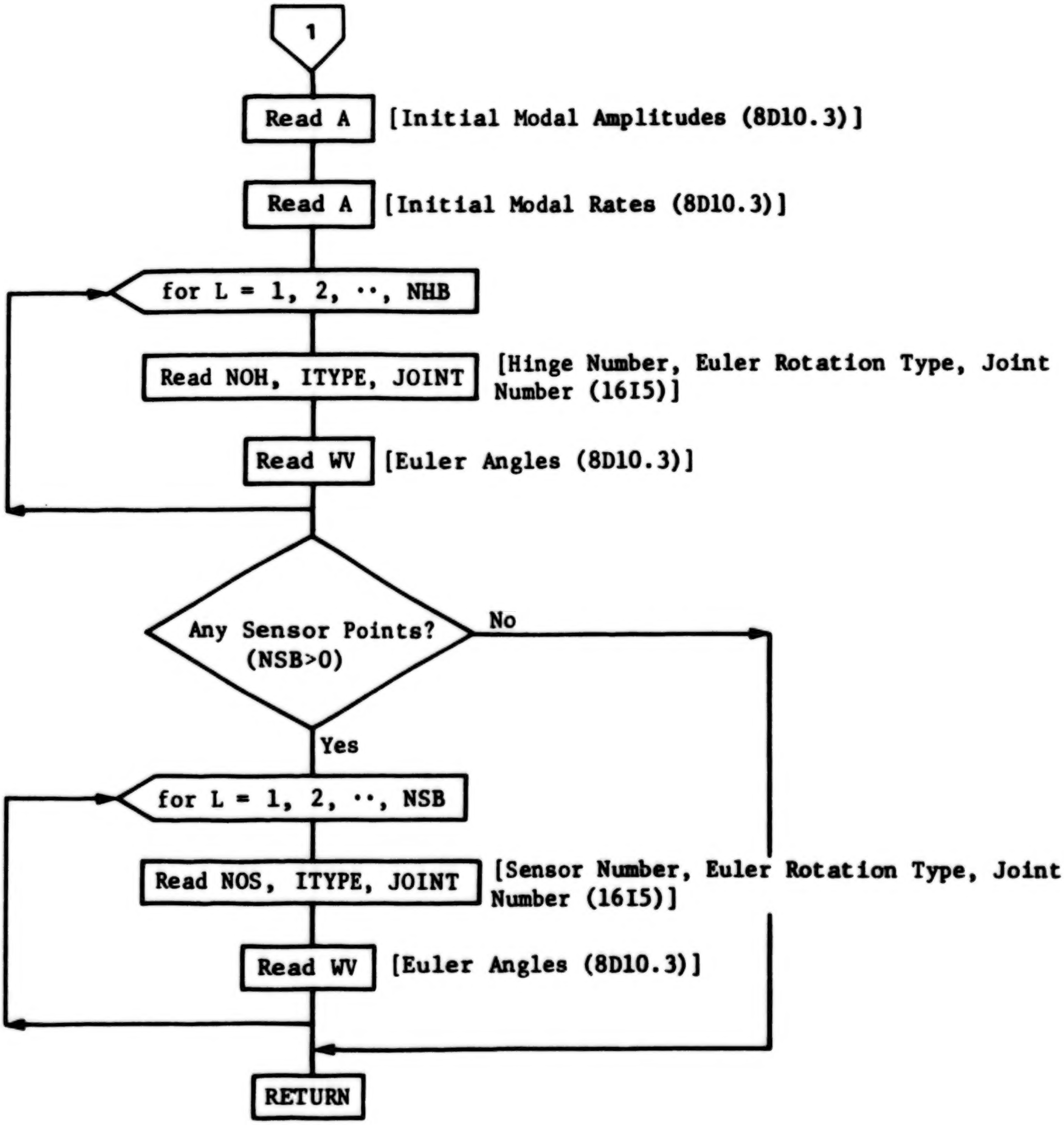

1
Read A
[Initial Modal Amplitudes (8D10.3)]
Read A
[Initial Modal Rates (8D10.3)]
for L = 1, 2, ··, NHB
Read NOH, ITYPE, JOINT
[Hinge Number, Euler Rotation Type, Joint Number (16I5)]
Read WV
[Euler Angles (8D10.3)]
Any Sensor Points? (NSB>0)
No
Yes
for L = 1, 2, ··, NSB
Read NOS, ITYPE, JOINT
[Sensor Number, Euler Rotation Type, Joint Number (16I5)]
Read WV
[Euler Angles (8D10.3)]
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 6 of 12).

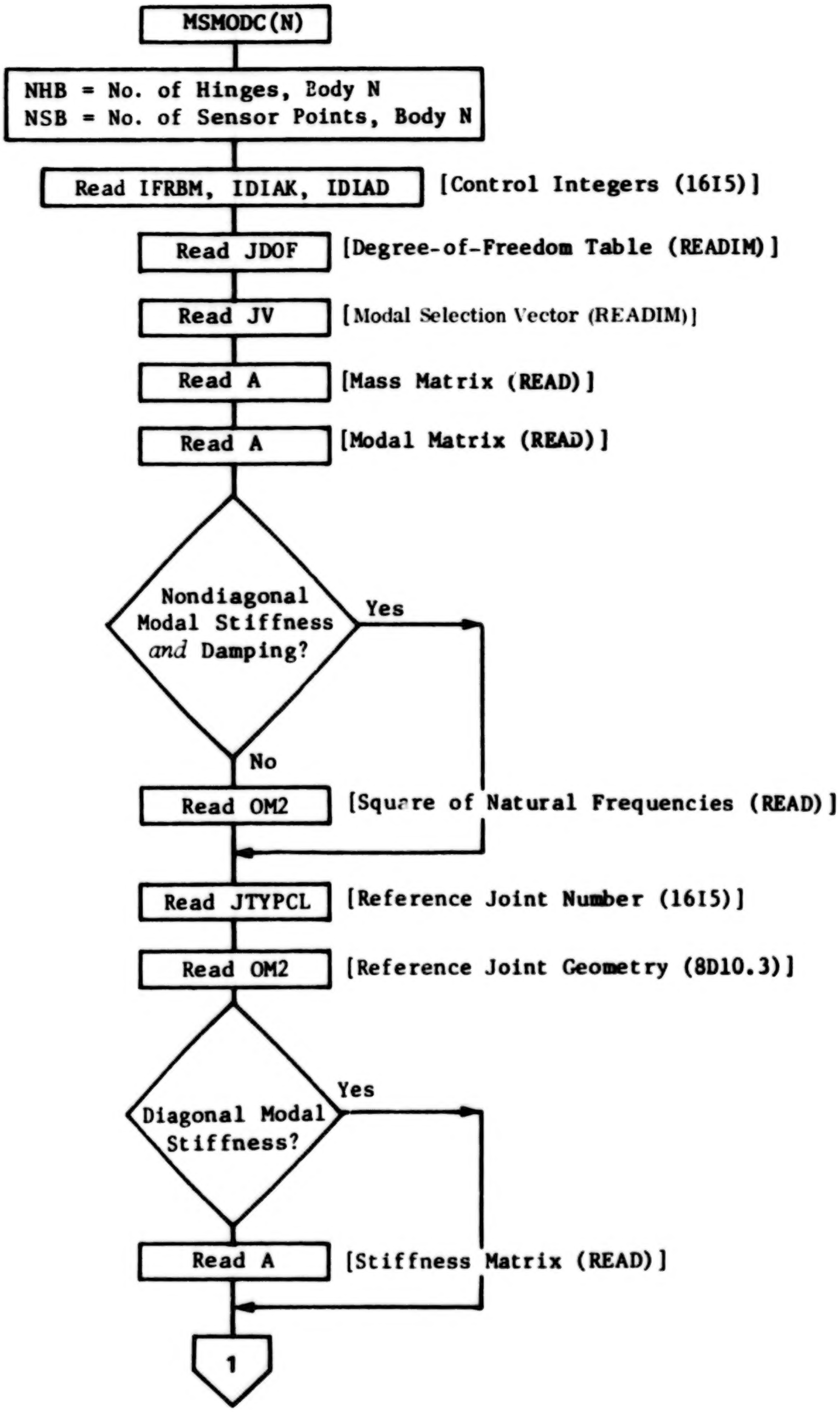

MSMODC(N)
NHB = No. of Hinges, Body N
NSB = No. of Sensor Points, Body N
Read IFRBM, IDIAK, IDIAD
[Control Integers (16I5)]
Read JDOF
[Degree-of-Freedom Table (READIM)]
Read JV
[Modal Selection Vector (READIM)]
Read A
[Mass Matrix (READ)]
Read A
[Modal Matrix (READ)]
Nondiagonal Modal Stiffness *and* Damping?
Yes
No
Read OM2
[Square of Natural Frequencies (READ)]
Read JTYPCL
[Reference Joint Number (16I5)]
Read OM2
[Reference Joint Geometry (8D10.3)]
Diagonal Modal Stiffness?
Yes
Read A
[Stiffness Matrix (READ)]
1

Figure 5. Program system DISCOS data stream flow (Sheet 7 of 12).

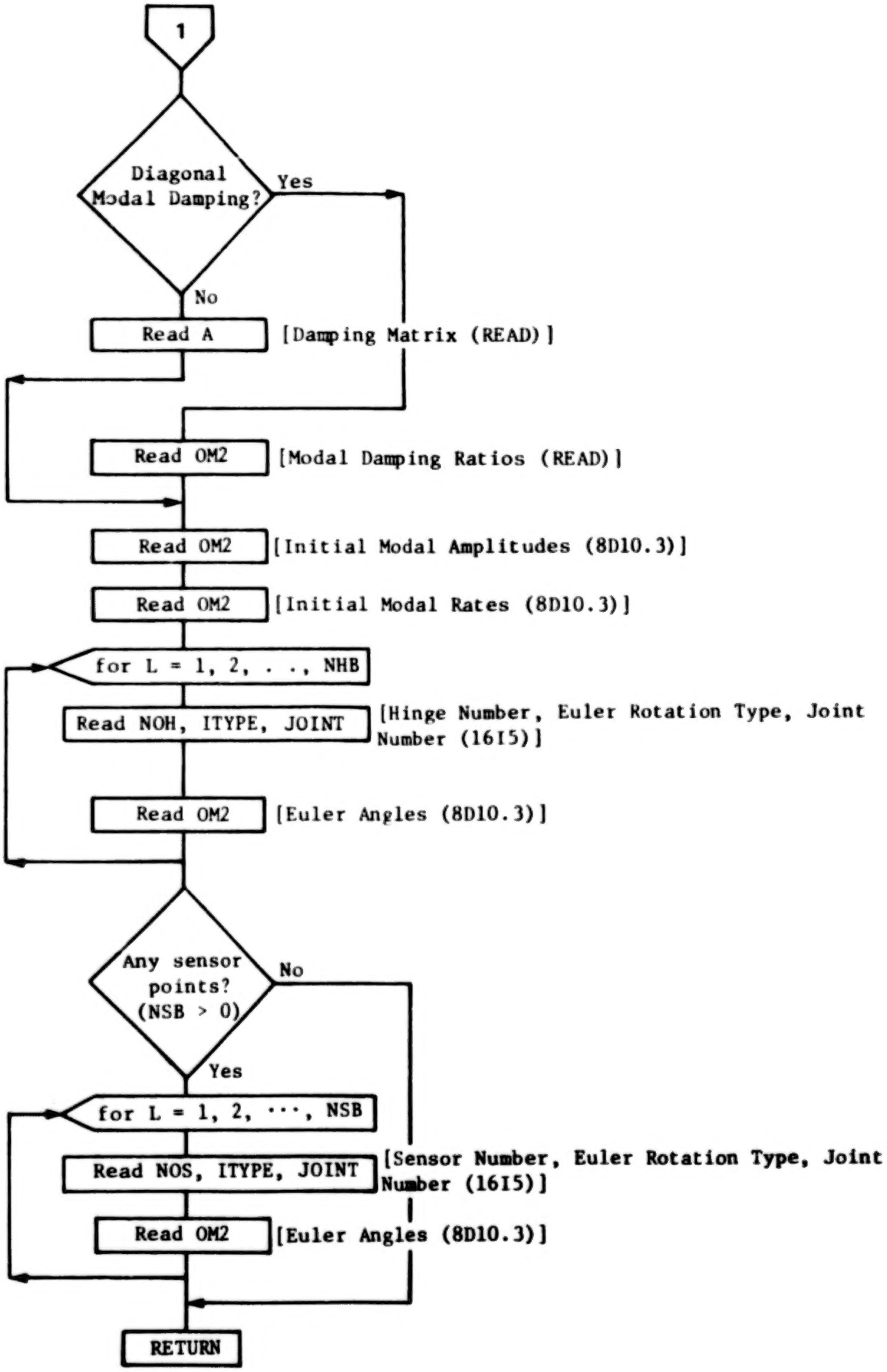

1
Diagonal Modal Damping?
Yes
No
Read A
[Damping Matrix (READ)]
Read OM2
[Modal Damping Ratios (READ)]
Read OM2
[Initial Modal Amplitudes (8D10.3)]
Read OM2
[Initial Modal Rates (8D10.3)]
for L = 1, 2, . . ., NHB
Read NOH, ITYPE, JOINT
[Hinge Number, Euler Rotation Type, Joint Number (16I5)]
Read OM2
[Euler Angles (8D10.3)]
Any sensor points? (NSB > 0)
No
Yes
for L = 1, 2, ···, NSB
Read NOS, ITYPE, JOINT
[Sensor Number, Euler Rotation Type, Joint Number (16I5)]
Read OM2
[Euler Angles (8D10.3)]
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 8 of 12).

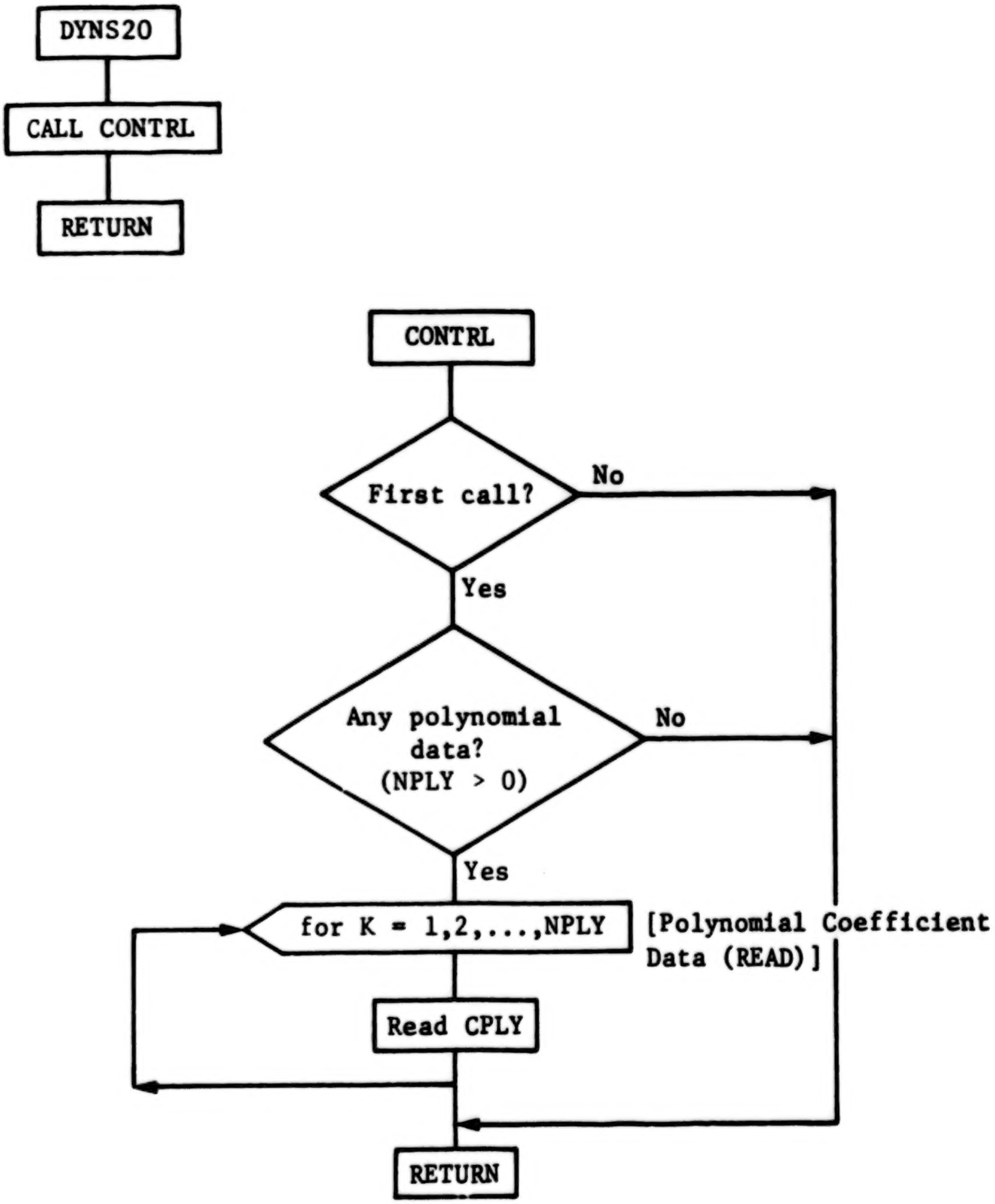

DYNS20
CALL CONTRL
RETURN
CONTRL
First call?
No
Yes
Any polynomial data? (NPLY > 0)
No
Yes
for K = 1,2,...,NPLY
[Polynomial Coefficient Data (READ)]
Read CPLY
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 9 of 12).

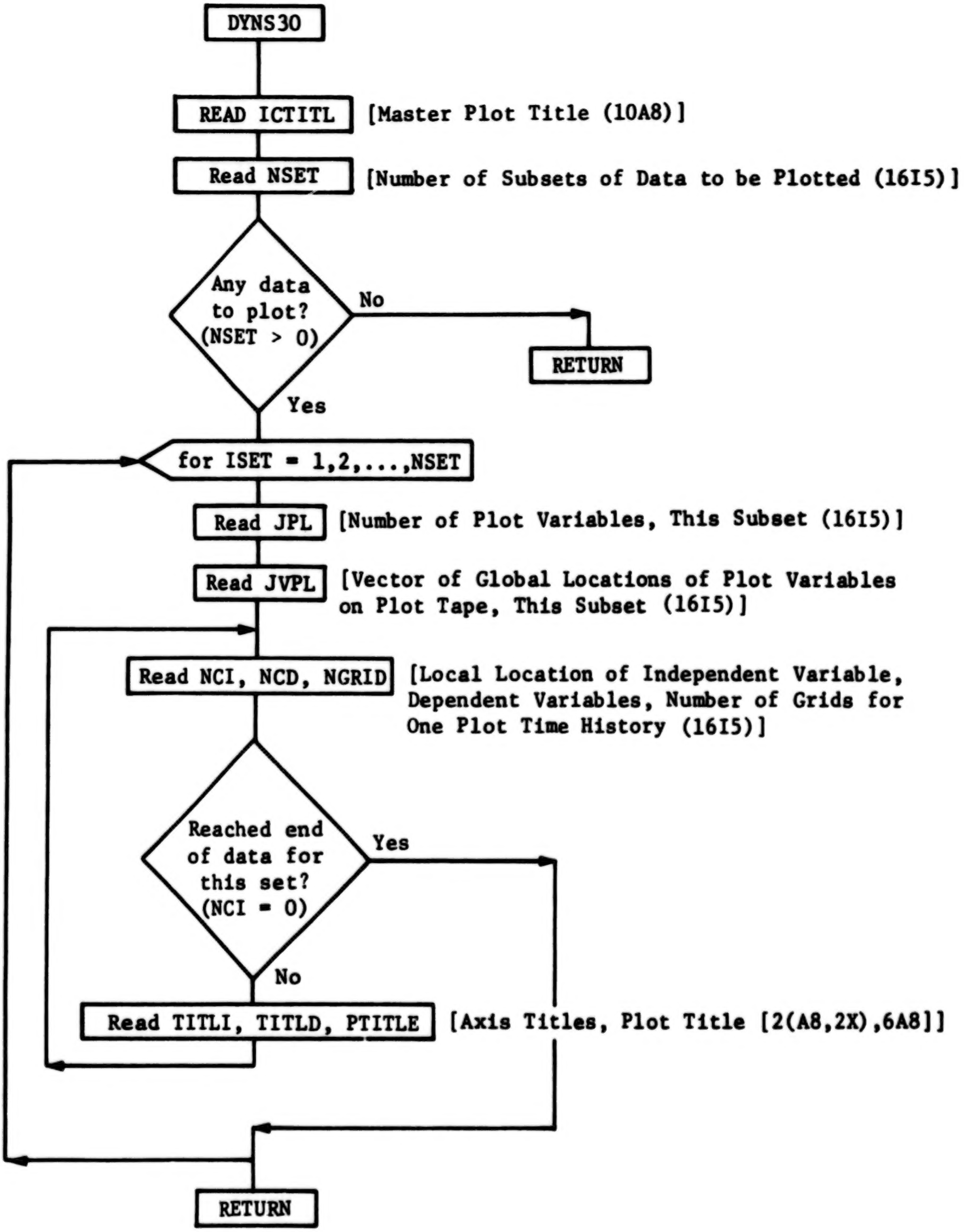

DYNS30
READ ICTITL
[Master Plot Title (10A8)]
Read NSET
[Number of Subsets of Data to be Plotted (16I5)]
Any data to plot? (NSET > 0)
No
RETURN
Yes
for ISET = 1,2,...,NSET
Read JPL
[Number of Plot Variables, This Subset (16I5)]
Read JVPL
[Vector of Global Locations of Plot Variables on Plot Tape, This Subset (16I5)]
Read NCI, NCD, NGRID
[Local Location of Independent Variable, Dependent Variables, Number of Grids for One Plot Time History (16I5)]
Reached end of data for this set? (NCI = 0)
Yes
No
Read TITLI, TITLD, PTITLE
[Axis Titles, Plot Title [2(A8,2X),6A8]]
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 10 of 12).

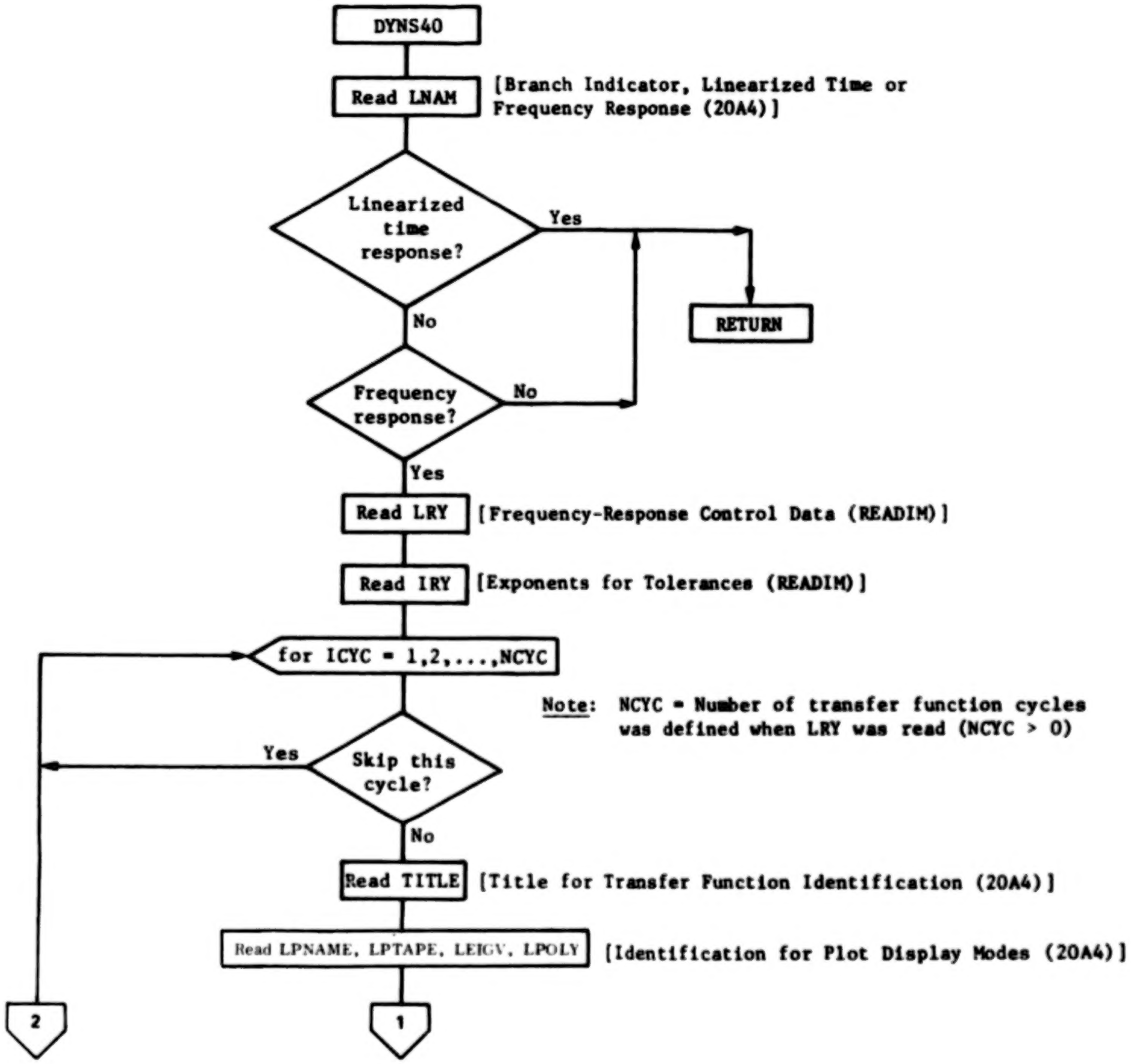

DYNS40
Read LNAM
[Branch Indicator, Linearized Time or Frequency Response (20A4)]
Linearized time response?
Yes
RETURN
No
Frequency response?
No
Yes
Read LRY
[Frequency-Response Control Data (READIM)]
Read IRY
[Exponents for Tolerances (READIM)]
for ICYC = 1,2,...,NCYC
Note: NCYC = Number of transfer function cycles was defined when LRY was read (NCYC > 0)
Yes
Skip this cycle?
No
Read TITLE
[Title for Transfer Function Identification (20A4)]
Read LPNAME, LPTAPE, LEIGV, LPOLY
[Identification for Plot Display Modes (20A4)]
2
1

Figure 5. Program system DISCOS data stream flow (Sheet 11 of 12).

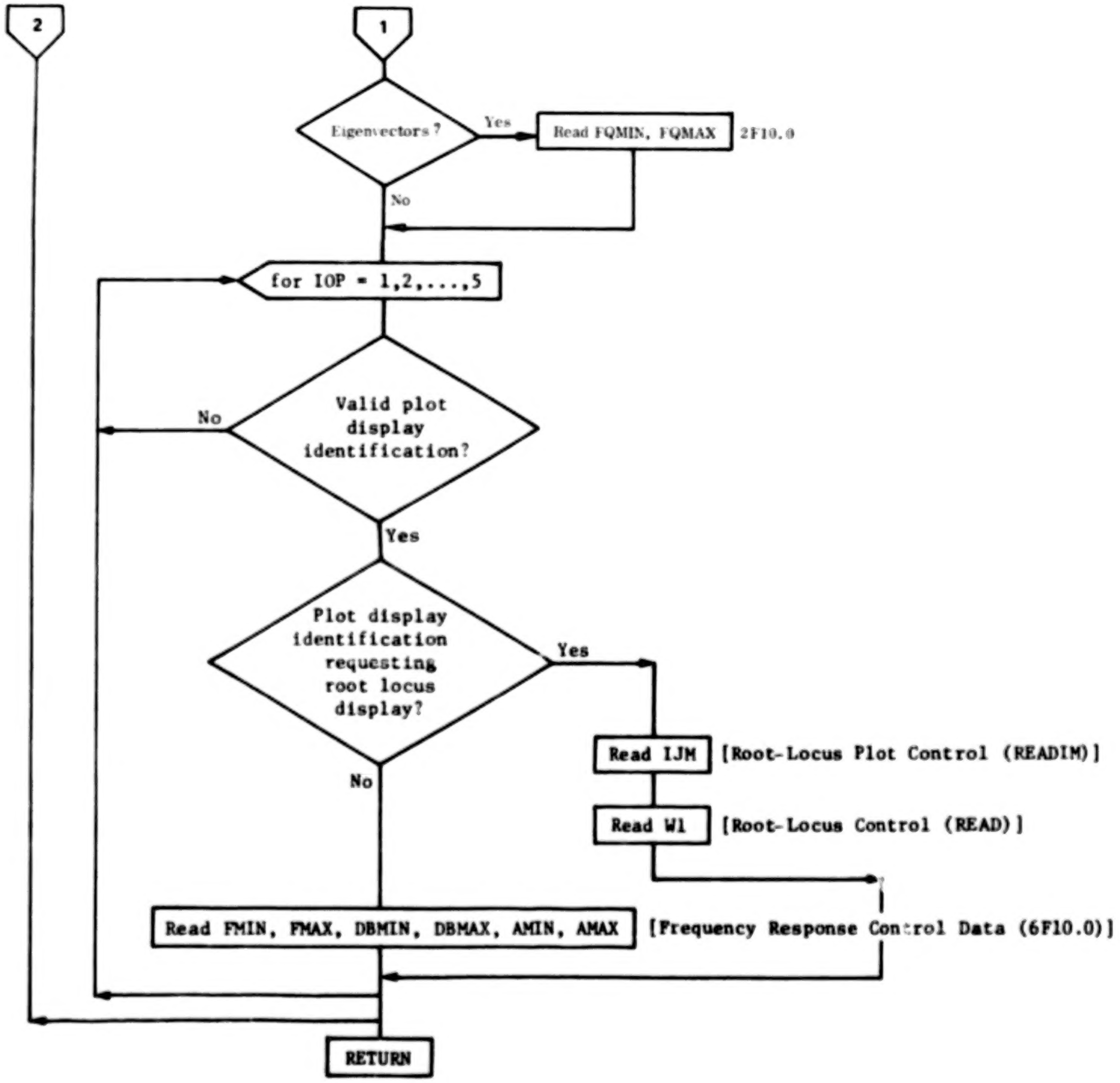

2
1
Eigenvectors ?
Yes
Read FQMIN, FQMAX
2F10.0
No
for IOP = 1,2,...,5
Valid plot display identification?
No
Yes
Plot display identification requesting root locus display?
Yes
Read IJM
[Root-Locus Plot Control (READIM)]
Read W1
[Root-Locus Control (READ)]
No
Read FMIN, FMAX, DBMIN, DBMAX, AMIN, AMAX
[Frequency Response Control Data (6F10.0)]
RETURN

Figure 5. Program system DISCOS data stream flow (Sheet 12 of 12).

much the same manner as the input data stream of Section IV to better acquaint the reader with the actual formatted output as it is presented by the program.

Note that the output stream will not reflect certain outputs that occur from routines that identify troublesome areas such as matrix singularities. Recall also that the basic input routines, READ and READIM, can also print out input matrix data as dictated by the user. These printouts will not be included. Refer to the theoretical volume (Volume I) and to the input data stream (Section IV) for correlating certain outputs with both the theory and the user input requirements.

Again, DISCOS documentation is intended to grow as new capability is added. The user is referred to subroutines DEF4 and DEF5 for a detailed description of the standard DISCOS output. In addition to the standard output, if any or all of the "DEBUG" flags are set by subroutine START input data, additional output data are obtained. The setting of these DEBUG flags for initial checkout work is encouraged. The associated output will provide the user with a detailed definition of the computation performed internally and resultant parametric magnitudes. The comment cards in subroutine DEF5 and the associated subroutines provide detailed definitions of parameters outputted.

VI. AUXILIARY PROGRAMS

This section describes three auxiliary digital codes that have been developed to aid the DISCOS program system user. The first code is a FORTRAN program that accepts the DISCOS code as input and, based on some additional user-supplied input, automatically redimensions the DISCOS source program to minimize core storage requirements. It can also be used to update the source tape with new versions of old subroutines. The other two digital codes are used to obtain NASTRAN-generated finite-element data, which define body flexibility, directly from NASTRAN and to process it into a DISCOS-acceptable format.

A. REDIM—The Redimension Program

The REDIM code was developed to aid the user in the efficient use of available digital-computer core-storage locations. Examination of existing digital-computer codes for generalized analyses of (possibly) large systems indicates that the nature of the code frequently dictates that a great deal of core-storage locations are required because of the sizes of program DIMENSION and COMMON blocks. This often leads to inefficient use of core storage because the user must have available sufficient core-storage locations to satisfy the program size. As a large percentage of program executions probably do not require the maximum dimension sizes of program storage blocks, it is obvious that an automatic procedure for altering the program code to meet specific requirements of a user would be desirable. Program REDIM was developed to satisfy these requirements.

REDIM is a self-contained code that contains an extensive list of format statements. The code reads the DISCOS source code from tape as coded data and reproduces it on the tape unless it finds an identifying format number in columns 73 through 75. In this case, it rewrites the source code according to the format corresponding to the identification. Therefore, REDIM provides an efficient and foolproof method of recasting the source input code to meet the user's requirements.

The following is a more detailed explanation of the manner in which the REDIM program can be used. Input data for the REDIM program are as follows:

- Data tapes:

  NIT = 2 = (Input tape) the data-set reference number associated with the DISCOS source code to be redimensioned and/or updated

  NOT = 1 = (Output tape) the data-set reference number associated with the DISCOS source code that has been redimensioned and/or updated

- Input Cards:

  Card 1 READ (5,109) NCHNG
  109 FORMAT (I5)

  NCHNG = Total number of subroutines on the input tape that will be totally updated

IF(NCHNG.EQ.O), this is the only data card required. The program immediately begins to read the source code from data set 2, to redimension it, and to output the redimensioned source on data set 1.

IF(NCHNG.GT.O), exactly NCHNG subroutines are to be updated. Because all subroutines are consecutively numbered by the DEBUG numbers, the subroutines to be updated are defined by the following input data cards:

READ(5,108) (NSUB(I),I=1),NCHNG)
108 FORMAT (16I5)

where

NSUB(I) = DEBUG number associated with the Ith-subroutine to be updated

All updated subroutines are read by the card reader immediately after the cards used to define the elements of NSUB. They are assumed to be in order of increasing DEBUG number. No separation cards are required between subroutines.

Subroutine DEF5 begins with a definition of the "maximum simulation model-size parameters." Each of these parameters can be altered in REDIM. The user is expected to simply change the appropriate variables in the REDIM source deck to accommodate specific redimensioning needs.

A listing of program REDIM follows:

```
CCC    PROGRAM TO DIMENSION DISCOS PROGRAM
CCCCC
C    REVISED NOV. 20, 1974.
CCCCC
C
       INTEGER IN(20)
       INTEGER NIN(120),NSUB(120)
      LOGICAL FLAG
      DIMENSION ICARD(18)
      DATA NIT, NOT / 2,  1 /
      DATA IEN, ID / 4H  EN, 4HD      /
C
      DATA  NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KY,JMAXC,JMAXL  /
     *          6,    6,    15,     5,     4,     12,250,   7,  500  /
      DATA   MAXZP,MXJVPL,MAXDUM   /
     *       1000,    16,  1500   /
      DATA MAXCNT   /
     *       150   /
      DATA L1,L5   / 99, 500   /
CCC
CCC   L1 MUST BE AN EVEN INTEGER.
CCC     IF NOT, IT WILL BE INCREMENTED BY 1
CCC     -- ---- -- --- -- ----------- -- ----
CCC
C
 1002 FORMAT (18A4,I3,I5)
  708 FORMAT (1X,18A4,I3,I5)
  703 FORMAT (1H1)
    1 FORMAT (5X1H*5X4HAMU(I2,1H,I2,1H,I2,5H),BW(I2,1H,I3,1H)37X,I3,I5)
    2 FORMAT (5X1H*5X5HBH(6,I2,1H,I2,7H),BS(6,I2,1H,I2,10H),ROL(3,3,I2,
     *   8H),DOL(3,I2,1H)16X,I3,I5)
    3 FORMAT (5X1H*5X4HGGS(I2,3H,9,I2,1H)49X,I3,I5)
    4 FORMAT (5X1H*5X7HHATH(3,I2,1H,I2,9H),SIGH(3,I2,1H,I2,9H),HATS(3,
     *I2,1H,I2,9H),SIGS(3,I2,1H,I2,1H)6X,I3,I5)
    5 FORMAT (5X1H*5X3HAM(I3,1H,I2,9H),ACOF(9,I2,1H,I2,
     *     9H),BCOF(6,I2,1H,I2,2H).22X,I3,I5)
    6 FORMAT (5X1H*5X6HCOF11(I2,1H,I2,1H,I2,8H),COF22(I2,1H,I2,1H,I2,
     *   8H),COF33(I2,1H,I2,1H,I2,5H),AK(I2,1H,I2,1H,I2,2H),I3,I5)
    7 FORMAT (5X1H*5X6HCOF12(I2,1H,I2,1H,I2,8H),COF13(I2,1H,I2,1H,I2,
     *   8H),COF23(I2,1H,I2,1H,I2,5H),AD(I2,1H,I2,1H,I2,2H),I3,I5)
    8 FORMAT (5X1H*5X6HCOFXY(I2,1H,I2,1H,I2,8H),COFXZ(I2,1H,I2,1H,I2,
     *   8H),COFYZ(I2,1H,I2,1H,I2,1H)14X,I3,I5)
    9 FORMAT (5X1H*5X5HIV(6,I2,1H)53X,I3,I5)
   10 FORMAT (5X1H*5X5HALAM(I2,1H)53X,I3,I5)
   11 FORMAT (5X1H*5X2HP(I3,7H),PMOM(I2,24H),HTOT(3),TOTL(3),ENGKE(I2,
     *   8H),ENGPE(I2,2H),9X,I3,I5)
   12 FORMAT (5X1H*5X6HNHPOI(I2,9H), NSPOI(I2,1H)41X,I3,I5)
   13 FORMAT (5X1H*5X8HPIN(3,3,I2,11H), RP2(3,3,I2,11H), RP3(3,3,I2,
     *   1H)24X,I3,I5)
   14 FORMAT (5X1H*5X3HPR(I3,3H,5)52X,I3,I5)
   15 FORMAT (5X1H*5X4HORK(I3,12H),PRK(4), NT42X,I3,I5)
   16 FORMAT (5X1H*5X8HBETAH(6,I2,11H),BETAHD(6,I2,8H),AMO(2,I2,
     *   9H),RH(3,3,I2,9H),RS(3,3,I2,2H),4X,I3,I5)
   17 FORMAT (5X1H*5X5HDH(3,I2,7H),DS(3,I2,8H),IMO(3,I2,
     *   7H),NMOW(I1,1H,I2,9H),IFTSMW(I2,2H),11X,I3,I5)
   18 FORMAT (5X1H*5X33HNB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2,I2,
     *   9H),IRGFLX(I2,11H),IHDATA(7,I2,2H),I3,I5)
   19 FORMAT (5X1H*5X5HLOCU(I2,7H),LENU(I2,11H),NU,NBETA,
     *   8HNLAM,NEQ26X,I3,I5)
```

```
20 FORMAT (5X1H*5X2HY(I3,6H),YDT(I3,1H)46X,I3,I5)
21 FORMAT (5X1H*5X6HINDEP(I3,1H)51X,I3,I5)
41 FORMAT (6X9HNBMAX  = I2,55X,I3,I5)
42 FORMAT (6X9HNHMAX  = I2,55X,I3,I5)
43 FORMAT (6X9HNSPMAX = I2,55X,I3,I5)
44 FORMAT (6X9HNMWMAX = I2,55X,I3,I5)
45 FORMAT (6X9HNMWBOD = I2,55X,I3,I5)
46 FORMAT (6X9HNMDBOD = I2,55X,I3,I5)
47 FORMAT (6X9HKMU    = I2,55X,I3,I5)
48 FORMAT (6X9HKY     = I3,54X,I3,I5)
49 FORMAT (6X9HKU     = I3,54X,I3,I5)
61 FORMAT (6X12HDIMENSION A(I3,1H,I3,4H),B(I3,1H,I3,5H),IV(I3,
         5H),JV(I3,19H),C(6,6),PHR6(6,6),1X,I3,I5)
62 FORMAT (5X1H*3X5HBC(9,I2,6H),WS1(I2,1H,I2,6H),WS2(I2,1H,I2,
  *     6H),OM2(I3,8H),OMGA2(I2,7H),JDOF(I2,3H,6)3X,I3,I5)
63 FORMAT (6X28HDATA NIT,NCT,KAB,KJDOF/ 5,6,I3,1H,I2,2H /30X,I3,I5)
64 FORMAT (6X12HDIMENSION A(I4,1H,I2,4H),B(I4,1H,I2,
  *     4H),C(I4,1H,I2,6H),AMU(I3,1H,I3,1H)11X,I3,I5)
65 FORMAT (6X13HDIMENSION WS(I4,1H,I2,7H),UVEC(I4,
  *     7H),WV(3)28X,I3,I5)
66 FORMAT (5X5H*    /5X,I3,1H,4X,I2,4H    /43X,I3,I5)
69 FORMAT (6X35HDIMENSION A(KAR,1), R(KAR,1), IVEC(I3,1H)27X,I3,I5)
70 FORMAT (6X31HDIMENSION AZ(KAZ,1), B(KB,1),W(I3,1H)31X,I3,I5)
71 FORMAT (6X13HIF (NRB .GT. I3,11H) GO TO 99939X,I3,I5)
72 FORMAT (6X31HDIMENSION AZ(KAZ,1), B(KB,1),W(I3,1H)31X,I3,I5)
73 FORMAT (6X11HIF (NRB.GT.I3,13H .OR. NCB.GT.I3,
  *     33H .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ)3X,I3,I5)
74 FORMAT (6X15HDIMENSION ICON(I2,8H), IVEC(I2,7H),JCON(I3,
  *     4H),R(I2,9H),JBIGST(I2,9H),JBIGFN(I2,1H)I3,I5)
75 FORMAT (6X14HDIMENSION GGV(I3,4H),D(I2,4H),V(I2,9H),BDTQ(6,I2,
  *     9H),BDTP(6,I2,1H)14X,I3,I5)
76 FORMAT (6X14HDIMENSION BMB(I2,1H,I2,7H),BM(6,I2,1H,I2,1H)34X,I3,
  *            I5)
77 FORMAT (6X33HEQUIVALENCE (BW(1),BMB(1)),  (BW(I4,8H),BM(1))21X,I3,
  *          I5)
78 FORMAT (6X13HDIMENSION FY(I3,6H,3),Z(I3,7H),ZNEW(I3,5H),IV(I3,
  *     1H)22X,I3,I5)
79 FORMAT (6X30HEQUIVALENCE (PR(1),FY(1)),(PR(I4,12H),Z(1)),(PR(I4,
  *     10H),ZNEW(1))6X,I3,I5)
80 FORMAT (6X29HDIMENSION A(KA,1),B(KA,1),CL(I3,1H)33X,I3,I5)
81 FORMAT (5X1H*8X8HITOPW(2,I2,8H),ITOP(2,I2,8H),NBODE(I2,1H)27X,I3,
  *          I5)
82 FORMAT (6X14HDIMENSION BMB(I2,1H,I2,7H),BM(6,I2,1H,I2,
  *     8H), ITOP(I2,1H,I2,5H),WR(I2,1H)14X,I3,I5)
83 FORMAT (6X15HDIMENSION RW(3,I2,5H),CW(I2,
  *     15H,3),VW(9),WV(6)27X,I3,I5)
84 FORMAT (6X40HDIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1),WR(I3,
  *     1H)22X,I3,I5)
85 FORMAT (6X40HDIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1),WR(I3,
  *     1H)22X,I3,I5)
86 FORMAT (6X44HDIMENSION A(6,1),B(KMU,1),C(6,1),IV(6,1),RW(I2,
  *     1H)19X,I3,I5)
87 FORMAT (6X13HDIMENSION CW(I2,9H,3),RW(3,I2,5H),V1(I2,
  *     17H),WSK(3,3),TSHFT(I2,2H),12X,I3,I5)
88 FORMAT (5X1H*5X6HTEX(6,I2,7H),ISPN(I2,10H),V(6),V2(I2,1H)
  *       31X,I3,I5)
89 FORMAT (6X13HDIMENSION WV(I2,1H)50X,I3,I5)
90 FORMAT (6X13HDIMENSION ZP(I4,1H,I2,6H),DUM(I4,7H),JVPL(I2,
  *     1H)26X,I3,I5)
91 FORMAT (6X20HDATA KRPLOT,KCPLOT /I4,1H,I2,1H/38X,I3,I5)
92 FORMAT (6X17HDIMENSION HNGT(6,I2,1H)46X,I3,I5)
```

```
   93 FORMAT (6X16HDIMENSION VCM(3,I2,10H),TIN(3,3,I2,27H),SYSCM(3),SYSI
     *N(3,3),XMAS(I2,1H)6X,I3,I5)
   95 FORMAT (5X1H*5X7HCNTDTA(I3,1H)50XI3,I5)
   96 FORMAT (6X8HKCONT = I3,55XI3,I5)
   97 FORMAT (6X13HDIMENSION VI(I4,5H),VD(I4,1H)39XI3,I5)
   98 FORMAT (5X1H*5X7HASUMRY(I2,11H,6),ISUMRY(I2,
     *     10H,3),KSUMRY29X,I3,I5)
   99 FORMAT (6X9HKSUMRY = I2,55X,I3,I5)
  100 FORMAT (5X1H*5X16HIFL1,IFL2,DRVEC(I3,1H)41X,I3,I5)
C
  410 FORMAT (6X,12HDIMENSION Q(I3,1H,,I3,6H), V1(I3,1H),37X,I3,I5)
  411 FORMAT (6X,29HEQUIVALENCE (Q(1),W1(1)), (Q(,I4,8H),W2(1)),25X,
     *        I3,I5)
  412  FORMAT (7X14HDIMENSION VS1(I3,6H),VS2(I3,1H)38X,I3,I5)           412
  414  FORMAT (7X,14HDIMENSION RRN(I3,6H),RIN(I3,6H),RRD(I3,6H),RID(    414
     *                                 I3,6H),R2R(I3,6H),R2I(I3,1H)2XI3,I5)
  415 FORMAT (6X,17HDIMENSION ROOTRE(I3,9H),ROOTIE(I3,4H),W(I3,
     *     9H,4),IROW(I3,3H,2),12X,I3,I5)
  416 FORMAT (6X,13HDIMENSION XR(I3,5H),XI(I3,5H),VR(I3,5H),VI(I3,1H),
     *     25X,I3,I5)
  418  FORMAT (7X16HDIMENSION IJM(2,I3,8H),LRY(9,I3,
     *           9H), IRY(3,I3,10H), KBKP(3),13X,I3,I5)                  418
  420  FORMAT (5X1H915X3HIV(I3,10H)  , JV  (I3,1H),31X,I3,I5)            420
  422  FORMAT (5X1HA15X5HFBRN(I3,8H), FBNC(I3,8H), FBRD(I3,8H), FBDC(    422
     *           I3,1H)9XI3,I5)
  424  FORMAT (5X1HB15X2HR(I3,11H)   , RX  (I3,1H)31XI3,I5)              424
  426  FORMAT (5X1HC15X5HV1  (I3,8H), V2  (I3,8H), V3  (I3,1H)20XI3,I5) 426
  428 FORMAT (5X1HD15X5HXV1 (I3,8H), XV2 (I3,8H), XV3 (I3,              428
     *          8H), XV4 (I3,1H)9XI3,I5)
  430 FORMAT (5X1HE15X5HY    (I3,8H), YD  (I3,1H)31XI3,I5)              430
  432 FORMAT (5X1HF15X3HW1(I3,1H,I3,6H), W2(I3,1H,I3,1H)27XI3,I5)       432
  434 FORMAT (6X17HEQUIVALENCE (VS1(I3,9H),VS2(1))37XI3,I5)              434
  436 FORMAT (6X6HKR  = I3,57XI3,I5)                                    436
  438 FORMAT (6X6HKRT = I3,57XI3,I5)                                    438
  440 FORMAT (6X6HKRX = I3,57XI3,I5)                                    440
  442 FORMAT (6X6HKV1 = I3,57XI3,I5)                                    442
  444 FORMAT (6X6HKV2 = I3,57XI3,I5)                                    444
  446 FORMAT (6X6HKVX = I3,57XI3,I5)                                    446
  447 FORMAT (5X1H*12X6HSAVED(I3,9H), SAVEP(I3,9H), SAVED(I3,           447
     *                  9H), SAVEA(I3,8H), KSAVE ,1XI3,I5)
  448 FORMAT (6X15HDIMENSION IVEC(I3,8H), JVEC(I3,1H)36XI3,I5)
  450 FORMAT (6X12HDIMENSION A(I3,4H),R(I3,4H),W(I3,4H),B(I3,4H),C(I3,  450
     *             5H),SS(I3,1H)14XI3,I5)
  452 FORMAT (6X14HDIMENSION PAR(I3,7H), QAR(I3,1H)38XI3,I5)            452
  454 FORMAT (6X16HDIMENSION XSAVE(I3,8H),YSAVE(I3,1H)35XI3,I5)         454
  456 FORMAT (5X12H1       6 ,  I3,4H   /,48XI3,I5)                     456
  460 FORMAT (6X14HDIMENSION PAR(I3,7H), QAR(I3,1H)38XI3,I5)            460
  462 FORMAT (5X1H110X15H TITLE(1) , WD(I3,
     1            25H) , TABG(20) , TABUP(31),13XI3,I5)
  464 FORMAT (5X1H310X5HFNUM(I3,8H), FDEN(I3,1H)36XI3,I5)               464
  466 FORMAT (6X9HDATA KWD/I3,1H/,53XI3,I5)
  468 FORMAT (6X12HDATA KDSAVE/I3,1H/50XI3,I5)
  470 FORMAT (6X22HDIMENSION PRK(4), YDS(I3,8H,3), YS(I3,1H)29XI3,I5)   470
  472 FORMAT (6X22HDIMENSION TITLE(1), X(I3,5H), Y(I3,1H)32XI3,I5)      472
  474 FORMAT (6X15HDIMENSION FAZE(I3,1H)7H,DBAMP(I3,1H)36XI3,I5)        474
  480 FORMAT (6X18HCOMMON /LWRKV1/ W(I3,1H)44XI3,I5)
C
      IGASMX = 150
      MLTAMX = 150
      MXBTAB = 150
      MLT3MX = 100
      MADMAX = 100
C
      IF (MOD(L1,2) .EQ. 1) L1 = L1 + 1
      L2 = 2*L1 + 7
      L3 = L1*L1/2  +  1
      L4 = 2*L2
      L2P1 = L2 + 1
      L01 = NBMAX
      L02 = NHMAX
      L03 = NSPMAX
      L04 = NMWMAX
      L05 = NMWBOD
      L06 = NMDBOD
      L07 = KY
      L08 = 6 + NMDBOD + NMWBOD
      L09 = 6*NHMAX
      L10 = NBMAX*(6 + NMDBOD) + NMWMAX
      L11 = 6 + NMDBOD
      L12 = 2*NHMAX - 1
      L13 = 2*(NHMAX - 1)
      L14 = 6*NBMAX
      L15 = 6*(NHMAX - 1)
      L16 = 7*(NHMAX - 1)
      L17 = 2*NSPMAX
      L18 = 2 + NMWBOD
      L19 = 2*NBMAX + 2
C
      L20 = L09*L09 + 1
      L40 = 25
      IF (L06 .GT. 12) L40 =2*L06+1
      L41 = JMAXC
      L42 = 6*JMAXC
      L43 = JMAXL
      L44 = 13
      L45 = NMDBOD
      L46 = IGASMX
      L47 = MLTAMX
      L48 = MXBTAB
      L49 = 3*L07+1
      L50 = 4*L07+1
      L51 = MLT3MX
      L52 = MADMAX
      L53 = MAXZP
      L54 = MXJVPL
      L55 = MAXDUM
      L56 = MAXCNT
      L57 = MAX0(9,NMDBOD)
      L58 = MAX0(10,NMDBOD)
      L59 = ((L06+6)*(L06+7))/2
      L60 = MAX0(L02,L03)
C
      KC1 = L09*L10
      KC2 = L09*L09+6*L08*L12
      IF (KC2 .GT. KC1) L10 = KC2/L09 + 1
C
CCCCCCCC
      REWIND NIT
CCCCCCCC
      DO 805 I=1,120
  805 NIN(I) = 2
      READ(5,109) NCHNG
      WRITE(6,209) NCHNG
  209 FORMAT (I5,' UPDATED SUBROUTINES WILL BE READ IN')
      IF(NCHNG.EQ.0) GO TO 812
      READ(5,108) (NSUB(I),I=1,NCHNG)
      WRITE(6,208) (NSUB(I),I=1,NCHNG)
  208 FORMAT (' DEBUG NUMBERS FOR THE SUBROUTINES ARE:',16I5)
      DO 806 I=1,NCHNG
      NN = NSUB(I)
  806 NIN(NN) = 5
  812 CONTINUE
      ICNT = 1
      M = 0
  500 CONTINUE
      M = M + 1
      NNN = NIN(ICNT)
      READ (NNN,1002,END=700) ICARD,N,MM
      IF(ICARD(2).EQ.IEN.AND.ICARD(3).EQ.ID) GO TO 810
      GO TO 808
  810 IF(NIN(ICNT).EQ.5) WRITE(NOT,1002) ICARD,N,M
  807 IF(NIN(ICNT).EQ.2) GO TO 809
      READ(2,1002,END=700) ICARD,N,MM
      IF(ICARD(2).EQ.IEN.AND.ICARD(3).EQ.ID) GO TO 811
      GO TO 807
  811 ICNT = ICNT + 1
      GO TO 500
  809 ICNT = ICNT + 1
C
  808 CONTINUE
      IF (N .GT. 0) GO TO 600
      WRITE (NOT,1002) ICARD,N,M
      GO TO 500
  600 IF (N .LT. 11) GO TO 601
      IF (N .LT. 21) GO TO 602
      IF (N .LT. 49) GO TO 603
      IF (N .LT. 70) GO TO 604
      IF (N .LT. 80) GO TO 605
      IF (N .LT.101) GO TO 606
      IF (N .LT. 428) GO TO 610
      IF (N .LT. 450) GO TO 612
      GO TO 615
  601 IF(N.EQ. 1)WRITE (NOT,1)  L08,L08,L01,L09,L10,N,M
      IF(N.EQ. 2)WRITE (NOT,2)  L11,L12,L11,L03,L01,L01,N,M
      IF(N.EQ. 3)WRITE (NOT,3)  L06,L01,N,M
      IF(N.EQ. 4)WRITE (NOT,4)  L06,L13,L06,L13,L06,L03,L06,L03,N,M
      IF(N.EQ. 5)WRITE (NOT,5)  L59,L01,L06,L01,L06,L01,N,M
      IF(N.EQ. 6)WRITE (NOT,6)  L06,L06,L01,L06,L06,L01,L06,L06,L01,
     *      L06,L06,L01,N,M
      IF(N.EQ. 7)WRITE (NOT,7)  L06,L06,L01,L06,L06,L01,L06,L06,L01,
     *      L06,L06,L01,N,M
      IF(N.EQ. 8)WRITE (NOT,8)  L06,L06,L01,L06,L06,L01,L06,L06,L01,N,M
      IF(N.EQ. 9)WRITE (NOT,9)  L02,N,M
      IF(N.EQ.10)WRITE (NOT,10) L09,N,M
      GO TO 500
  602 IF(N.EQ.11)WRITE (NOT,11) L10,L14,L01,L01,N,M
      IF(N.EQ.12)WRITE (NOT,12) L01,L01,N,M
      IF(N.EQ.13)WRITE (NOT,13) L02,L02,L02,N,M
      IF(N.EQ.14)WRITE (NOT,14) L07,N,M
      IF(N.EQ.15)WRITE (NOT,15) L07,N,M
      IF(N.EQ.16)WRITE (NOT,16) L02,L02,L04,L15,L17,N,M
      IF(N.EQ.17)WRITE (NOT,17) L16,L17,L04,L18,L01,L03,N,M
      IF(N.EQ.18)WRITE (NOT,18) L02,L01,L02,N,M
      IF(N.EQ.19)WRITE (NOT,19) L19,L19,N,M
      IF(N.EQ.20)WRITE (NOT,20) L07,L07,N,M
      GO TO 500
```

```
603 IF(N.EQ.21)WRITE (NOT,21) L07,N,M
    IF(N.EQ.41)WRITE (NOT,41)  L01,N,M
    IF(N.EQ.42)WRITE (NOT,42)  L02,N,M
    IF(N.EQ.43)WRITE (NOT,43)  L03,N,M
    IF(N.EQ.44)WRITE (NOT,44) L04,N,M
    IF(N.EQ.45)WRITE (NOT,45)  L05,N,M
    IF(N.EQ.46)WRITE (NOT,46)  L06,N,M
    IF(N.EQ.47)WRITE (NOT,47)  L08,N,M
    IF(N.EQ.48)WRITE (NOT,48) L07,N,M
    GO TO 500
604 IF(N.EQ.49)WRITE (NOT,49)  L10,N,M
    IF(N.EQ.61)WRITE (NOT,61)   L42,L42,L42,L42,L42,L42,N,M
    IF(N.EQ.62)WRITE (NOT,62)  L06,L06,L06,L06,L06,L42,L11,L41,N,M
    IF(N.EQ.63)WRITE (NOT,63)  L42,L41,N,M
    IF(N.EQ.64)WRITE(NOT,64)  L43,L45,L43,L45,L43,L45,L11,L11,N,M
    IF(N.EQ.65)WRITE(NOT,65)  L43,L44,L43,N,M
    IF(N.EQ.66)WRITE(NOT,66)  L43,L45,N,M
    IF(N.EQ.69)WRITE (NOT,69)  L46,N,M
    GO TO 500
605 IF(N.EQ.70)WRITE (NOT,70)  L47,N,M
    IF(N.EQ.71)WRITE (NOT,71)  L47,N,M
    IF(N.EQ.72)WRITE (NOT,72)  L48,N,M
    IF(N.EQ.73)WRITE (NOT,73)  L48,L48,N,M
    IF(N.EQ.74)WRITE (NOT,74)  L09,L09,L10,L09,L02,L02,N,M
    IF(N.EQ.75)WRITE (NOT,75)  L10,L09,L09,L11,L11,N,M
    IF(N.EQ.76)WRITE (NOT,76)  L09,L09,L08,L12,N,M
    IF(N.EQ.77)WRITE (NOT,77)   L20,N,M
    IF(N.EQ.78)WRITE (NOT,78)  L07,L07,L07,L07,N,M
    IF(N.EQ.79)WRITE (NOT,79)  L49,L50,N,M
    GO TO 500
606 IF(N.EQ.80)WRITE (NOT,80)  L08,N,M
    IF(N.EQ.81)WRITE (NOT,81)  L02,L02,L01,N,M
    IF(N.EQ.82)WRITE (NOT,82)  L09,L09,L08,L12,L02,L01,L11,N,M
    IF(N.EQ.83)WRITE (NOT,83)  L06,L06,N,M
    IF(N.EQ.84)WRITE (NOT,84)  L51,N,M
    IF(N.EQ.85)WRITE (NOT,85)  L52,N,M
    IF(N.EQ.86)WRITE (NOT,86)  L11,N,M
    IF(N.EQ.87)WRITE (NOT,87)  L06,L06,L06,L04,N,M
    IF(N.EQ.88)WRITE (NOT,88)  L03,L03,L06,N,M
    IF(N.EQ.89)WRITE (NOT,89)  L06,N,M
    IF(N.EQ.90)WRITE (NOT,90)  L53,L54, L55,L54,N,M
    IF(N.EQ.91)WRITE (NOT,91)  L53,L54,N,M
    IF(N.EQ.92)WRITE (NOT,92) L02,N,M
    IF(N.EQ.93)WRITE(NOT,93) L01,L01,L01,N,M
    IF(N.EQ.95)WRITE (NOT,95)  L56,N,M
    IF(N.EQ.96)WRITE (NOT,96)  L56,N,M
    IF(N.EQ.97)WRITE (NOT,97)  L53,L53,N,M
    IF(N.EQ.98)WRITE(NOT,98)  L60,L60,N,M
    IF(N.EQ.99)WRITE(NOT,99)  L60,N,M
    IF (N.EQ.100)WRITE(NOT,100)  L46,N,M
    GO TO 500
610 CONTINUE
    IF (N .EQ. 410) WRITE (NOT,410) L1,L1,L1,N,M
    IF (N .EQ. 411) WRITE (NOT,411) L3,N,M
    IF (N .EQ. 412)  WRITE(NOT,412) L4,L2,N,M
    IF (N .EQ. 414)  WRITE (NOT,414)  (L1,I=1,6),N,M
   IF(N.EQ.415)WRITE (NOT,415) L1,L1,L1,L1,N,M
   IF(N.EQ.416)WRITE (NOT,416) L1,L1,L1,L1,N,M
    IF (N .EQ. 418)  WRITE (NOT,418) L1,L1,L1,N,M
    IF (N .EQ. 420)  WRITE (NOT,420) L07,L07,N,M
    IF (N .EQ. 422)  WRITE (NOT,422) (L1,I=1,4),N,M
    IF (N .EQ. 424)  WRITE (NOT,424) L2,L4,N,M
```

```
      IF (N .EQ. 426)  WRITE (NOT,426) L1,L1,L1,N,M
      GO TO 500
  612 CONTINUE
      IF (N .EQ. 428) WRITE (NOT,428) (L1,I=1,4),N,M
      IF (N .EQ. 430) WRITE (NOT,430) L07,L07,N,M
      IF (N .EQ. 432) WRITE (NOT,432) (L1,I=1,4),N,M
      IF (N .EQ. 434) WRITE (NOT,434) L2P1,N,M
      IF (N .EQ. 436) WRITE (NOT,436) L1,N,M
      IF (N .EQ. 438) WRITE (NOT,438) L2,N,M
      IF (N .EQ. 440) WRITE (NOT,440) L4,N,M
      IF (N .EQ. 442) WRITE (NOT,442) L4,N,M
      IF (N .EQ. 444) WRITE (NOT,444) L2,N,M
      IF (N .EQ. 446) WRITE (NOT,446) L1,N,M
      IF (N .EQ. 447) WRITE (NOT,447) (L5,I=1,4),N,M
      IF (N .EQ. 448) WRITE (NOT,448) L2,L2,N,M
      GO TO 500
  615 CONTINUE
      IF (N .EQ. 450) WRITE (NOT,450) (L1,I=1,6),N,M
      IF (N .EQ. 452) WRITE (NOT,452) L1, L1, N, M
      IF (N .EQ. 454) WRITE (NOT,454) L5, L5, N, M
      IF (N .EQ. 456) WRITE (NOT,456) L5,N,M
      IF (N .EQ. 460) WRITE (NOT,460) L1,L1,N,M
      IF (N .EQ. 462) WRITE (NOT,462) L2,N,M
      IF (N .EQ. 464) WRITE (NOT,464) L1,L1,N,M
      IF (N .EQ. 466) WRITE (NOT,466) L2,N,M
      IF (N .EQ. 468) WRITE (NOT,468) L5,N,M
      IF (N .EQ. 470) WRITE (NOT,470) L07,L07,N,M
      IF (N. EQ. 472) WRITE (NOT,472) L5, L5, N, M
      IF (N .EQ. 474) WRITE (NOT,474) L5,L5,N,M
      IF (N .EQ. 480) WRITE (NOT,480) L1,N,M
      GO TO 500
C
  700 REWIND 1
      IIC = 1
      WRITE(6,704) IIC
      IIC = 2
  704 FORMAT (////,90X,' SUBROUTINE',I5)
  706 READ (1,1002,END=707) ICARD,N,M
      WRITE (6,708) ICARD,N,M
      IF (ICARD(2).EQ.IEN .AND. ICARD(3).EQ.ID) GO TO 709
      GO TO 706
  709 WRITE(6,704) IIC
      IIC = IIC+1
      GO TO 706
  707 CONTINUE
      REWIND 1
      STOP
  101 FORMAT(20A4)
  102 FORMAT(18A4,I8)
  103 FORMAT(1X,A4,68X,I8)
  104 FORMAT(1X,18A4,I8)
  105 FORMAT('1')
  106 FORMAT(A4,68X,I8)
  107 FORMAT (80X,12A4)
  108 FORMAT(16I5)
  109 FORMAT(I5)
      END
```

B. Support Programs for the DISCOS Flexible-Body Capability

The flexible-body option available in the DISCOS program requires that the user obtain mode-dependent parameters. For most applications, these parameters will be computed from lumped parameter data obtained by the standard in-house finite-element program. Although NASTRAN is the standard Goddard Space Flight Center (GSFC), National Aeronautics and Space Administration (NASA), program, other installations make use of other programs. Therefore, the following two programs, submitted as a guide for the interested user, can either be used directly or be extensively modified to suit the situation at the particular computer installation at which DISCOS will be applied.

The first program is actually the NASTRAN Executive Control Deck for a specially written DMAP program. This DMAP program will output all lumped parameter data that are necessary for generating the required DISCOS input data. The N-BOD2 program referred to in the DMAP listing is another flexible-body program used at GSFC. It requires similar data but in a completely different format.

This DMAP program is operational on the Macneal-Schwendler version 38 of NASTRAN and the Level 15.5 NASA version with one small change in the functional module, GP4. (Built-in NASTRAN error messages will point out the change.)

The DMAP program was written so that, in addition to modes and frequencies (real), NASTRAN could be forced to output modal mass, stiffness, and damping matrices. Several print and punch options were also built in so that the user could see more clearly some of the resultant mass stiffness and damping matrices internally generated by NASTRAN. For a detailed description of the DMAP program, the interested user is referred to a NASTRAN Level 15 User's Manual* and Programmer's Manual.† It is basically a combination of rigid formats 3, 9, and 12.

```
  ID      FRISCH, DISCOS N-BOD2 NASTRAN
  DIAG    14
TIME     6000
  APP     DMAP
  BEGIN   DISCOS AND N-BOD2 DATA VIA NASTRAN $
  FILE    $
  GP1     GEOM1,GEOM2,/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/ C,N,123/
          V,N,NOGPDT $
  SAVE    LUSET,NOGPDT $
  CHKPNT GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL  $
  PARAM   //C,N,NOP/V,Y,PRTGD=1 $
  COND    LBL10,PRTGD $
  TABPT      GPL,EQEXIN,GPDT,BGPDT,SIL// $
  TABPRT GPL//C,N,GPL $
  TABPRT GPDT//C,N,GPDT $
  TABPRT BGPDT//C,N,BGPDT $
  LABEL   LBL10 $
```

---

*Calab W. McCormick, *The NASTRAN User's Manual,* NASA SP-222, September 1970.

†Frank J. Douglas, *The NASTRAN Programmer's Manual,* NASA SP-223, September 1970.

```
PURGE   USET,GM,GO,KAA,BAA,MAA,K4AA,PST,KFS,QP,EST/NOGPDT $
CHKPNT  USET,GM,GO,KAA,BAA,MAA,K4AA,PST,KFS,QP,EST   $
GP2     GEOM2,EQEXIN/ECT $
CHKPNT  ECT $
TA1,    ,ECT,EPT,BGPDT,SIL,,CSTM/EST,,GEI,ECPT,GPCT/V,N,LUSET/ C,N,
        123/V,N,NOSIMP/C,N,0/V,N,NOGENL/V,N,GENEL $
SAVE    NOGENL,NOSIMP,GENEL $
CHKPNT  EST,GEI,ECPT,GPCT $
PURGE   K4GG,GPST,OGPST,MGG,BGG, K4NN,K4FF,K4AA,MNN,MFF,MAA,BNN,BFF,
        BAA,KGGX/NOSIMP/ OGPST/GENEL $
CHKPNT  K4GG,GPST,OGPST,MGG,BGG,K4NN,K4FF,K4AA,MNN,MFF,MAA,BNN,BFF,
        BAA,KGGX   $
SMA1    CSTM,MPT,ECPT,GPCT, DIT/KGGX,K4GG,GPST/V,N,NOGENL/ V,N,NOK4GG$
SAVE    NOK4GG $
CHKPNT  KGGX,K4GG,GPST $
PARAM   //C,N,NOP/V,Y,PRTKG=-1 $
COND    LBL14,PRTKG $
MATPRN  KGGX,,,,// $
LABEL   LBL14 $
PURGE   K4NN,K4GG,K4FF,K4AA/NOK4GG $
CHKPNT  K4NN,K4GG,K4FF,K4AA   $
SMA2    CSTM,MPT,ECPT,GPCT,DIT/MGG,BGG/V,Y,WTMASS=1.0/V,N,NOMGG/ V,N,
        NOBGG/V,Y,COUPMASS=-1 $
SAVE    NOMGG,NOBGG $
CHKPNT  MGG,BGG   $
PARAM   //C,N,NOP/V,Y,PRTMG=-1   $
COND    LBL11,PRTMG $
MATPRN   MGG,,,,// $
LABEL   LBL11 $
PARAM   //C,N,NOP/V,Y,PRTMGD=1 $
COND    LBL15,PRTMGD $
DIAGONAL MGG/MMG/C,Y,OPT=COLUMN/V,Y,POWER=1. $
MATPRN  MMG,,,,// $
LABEL   LBL15 $
PURGE   BNN,BFF,BAA,BGG/NOBGG $
CHKPNT  BNN,BFF,BAA,BGG $
COND    LBL1,GRDPNT $
GPWG    BGPDT,CSTM,EQEXIN,MGG/OGPWG/V,Y,GRDPNT=-1/V,Y,WTMASS $
CHKPNT  OGPWG   $
OFP     OGPWG,,,,,//V,N,CARDNO $
SAVE    CARDNO $
LABEL   LBL1 $
PARAM   //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 $
GP4     CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPDT,CSTM/RG,,USET,/V,N,LUSET/
        V,N,MPCF1/V,N,MPCF2/V,N,SINGLE/V,N,OMIT/V,N,REACT/V,N,NSKIP/
        V,N,REPEAT/V,N,NOSET/V,N,NOL/V,N,NOA/C,N,0 $
SAVE    MPCF1,SINGLE,OMIT,NOSET,REACT,MPCF2,NSKIP,REPEAT,NOL,NOA $
CHKPNT  RG,USET $
PURGE   GM,GMD/MPCF1/GO,KOOB,LOO,UOO,MOOB,MOAB,GOD/OMIT/KFS,PST,QP/
        SINGLE $
CHKPNT  GM,GMD,GO,KOOB,LOO,UOO,MOOB,MOAB,GOD,KFS,PST,QP $
EQUIV   KGGX,KNN/MPCF1/MGG,MNN/MPCF1/ BGG,BNN/MPCF1/K4GG,K4NN/MPCF1 $
CHKPNT  KNN,MNN,BNN,K4NN,OGPST $
GPSP    GPL,GPST,USET,SIL/OGPST $
OFP     OGPST,,,,,//V,N,CARDNO $
SAVE    CARDNO $
COND    LBL2,MPCF1 $
MCE1    USET,RG/GM $
MCE2    USET,GM,KGGX,MGG,BGG,K4GG/KNN,MNN,BNN,K4NN $
```

```
CHKPNT GM,KNN,MNN,BNN,K4NN $
LABEL  LBL2 $
EQUIV  KNN,KFF/SINGLE/MNN,MFF/SINGLE/BNN,BFF/SINGLE/K4NN,K4FF/SINGLE $
CHKPNT KFF,MFF,BFF,K4FF  $
COND   LBL3,SINGLE $
SCE1   USET,KNN,MNN,BNN,K4NN/KFF,KFS, ,MFF,BFF,K4FF $
LABEL  LBL3 $
EQUIV  KFF,KAA/OMIT/ MFF,MAA/OMIT/BFF,BAA/OMIT/K4FF,K4AA/OMIT $
CHKPNT KAA,MAA,BAA,K4AA  $
COND   LBL20,OMIT $
SMP1   USET,KFF,,BFF,K4FF/GO,KAA,KOOB,LOO,UOO,,,,BAA,K4AA $
CHKPNT GO,KAA,KOOB,LOO,UOO,BAA,K4AA $
SMP2   USET,GO,MFF/MAA $
CHKPNT MAA $
LABEL  LBL20 $
PURGE  KRR,KLR,DM,MLR,MR/REACT/CM/MPCF1/GO/OMIT/KFS/SINGLE/QG/NOSET $
COND   LBL21,REACT $
RBMG1  USET,KAA,MAA/KLL,KLR,KRR,MLL,MLR,MRR $
CHKPNT KLL,KLR,KRR,MLL,MLR,MRR $
RBMG2  KLL/LLL,ULL $
CHKPNT ULL,LLL $
RBMG3  LLL,ULL,KLR,KRR/DM $
CHKPNT DM $
RBMG4  DM,MLL,MLR,MRR/MR $
CHKPNT MR $
LABEL  LBL21 $
DPD    DYNAMICS,GPL,SIL,USET/GPLD,SILD,USETD,,,,,,,EED,EQDYN/V,N,
       LUSET/V,N,LUSETD/V,N,NOTFL/V,N,NODLT/V,N,NOPSDL/V,N,NOFRL/V,N,
       NOMLFT/V,N,NOTRL/V,N,NOEED/C,N,123/V,N,NOUE $
SAVE   NOEED,NOUE $
CHKPNT GPLD,SILD,USETD,EED,EQDYN $
COND   ERRO2,NOEED
READ   KAA,MAA,MR,DM,EED,USET,CASECC/LAMA,PHIA,MI,OEIGS/C,N,MODES/ V,
       N,NEIGV $
SAVE   NEIGV$
CHKPNT LAMA,PHIA,MI,OEIGS $
ADD    MI,/OI/C,N,(1.-10,0.0) $
OFP    LAMA,OEIGS,,,,//V,N,CARDNO $
COND   FINIS,NEIGV $
SDR1   USET,,PHIA,,,GO,GM,,KFS,,/PHIG,,QG/C,N,1/C,N,REIG $
CHKPNT PHIG,QG  $
PARAM  //C,N,NOP/V,Y,PRTMD=1 $
COND   LBL12,PRTMD $
MATPRN    PHIG,,,,// $
LABEL  LBL12 $
SDR2   CASECC,CSTM,MPT,DIT,EQEXIN,SIL,,,BGPDT,LAMA,QG,PHIG,EST,/,OQG1,
       OPHIG,OES1,OEF1,PPHIG/C,N,REIG $
CHKPNT OQG1,OPHIG,OES1,OEF1,PPHIG $
OFP    OPHIG,OQG1,OEF1,OES1,,//V,N,CARDNO $
SAVE   CARDNO $
EQUIV  GO,GOD/NOUE/GM,GMD/NOUE $
CHKPNT GOD,GMD $
MTRXIN CASECC,MATPOOL,EQDYN,,/K2PP,M2PP,B2PP/V,N,LUSETD/V,N,NOK2PP/V,
       N,NOM2PP/V,N,NOB2PP $
SAVE   NOK2PP,NOM2PP,NOB2PP $
CHKPNT K2PP,M2PP,B2PP $
PARAM  //C,N,AND/V,N,KDEKA/V,N,NOUE/V,N,NOK2PP $
PARAM  //C,N,AND/V,N,MDEMA/V,N,NOUE/V,N,NOM2PP $
PARAM  //C,N,AND/V,N,KDEK2/V,N,NOGENL/V,N,NOSIMP $
```

```
   PURGE   K2DD/NOK2PP/M2DD/NOM2PP/B2DD/NOB2PP $
   EQUIV   M2PP,M2DD/NOA/B2PP,B2DD/NOA/K2PP,K2DD/NOA/MAA,MDD/MDEMA/ KAA,
           KDD/KDEKA
   GKAD    USETD,GM,GO,KAA,BAA,MAA,K4AA,K2PP,M2PP,B2PP/KDD,BDD,MDD,GMD,
           GOD,K2DD,M2DD,B2DD/C,N,TRANRESP/C,N,DISP/C,N,DIRECT/C,Y,G=0.0/
           C,Y,W3=0.0/C,Y,W4=0.0/V,N,NOK2PP/V,N,NOM2PP/V,N,NOB2PP/ V,N,
           MPCF1/V,N,SINGLE/V,N,OMIT/V,N,NOUE/V,N,NOK4GG/V,N,NOBGG/ V,N,
           KDEK2/C,N,-1 $
   CHKPNT M2DD,B2DD,K2DD,MDD,KDD,BDD,GMD,GOD $
   GKAM    USETD,PHIA,OI,LAMA,DIT,MDD,BDD,KDD,CASECC/MHH,BHH,KHH,
           PHIDH/V,N,NOUE/C,Y,LMODES=0/C,Y,LFREQ=0.0/ C,Y,HFREQ=0.0/C,N,
           1/C,N,1/C,N,1/V,N,NONCPU/V,N,FMODE $
   CHKPNT MHH,BHH,KHH,PHIDH  $
   PARAM   //C,N,NOP/V,Y,PRTMM=1 $
   COND    LBL13,PRTMM $
   MATPRN KHH,BHH,MHH,PHIDH,// $
   LABEL   LBL13 $
   PARAM //C,N,NOP/V,Y,PCHGD=-1 $
   COND    LBL6,PCHGD $
   TABPCH GPL,BGPDT,,,//C,N,GP/C,N,BG $
   LABEL   LBL6 $
   PARAM //C,N,NOP/V,Y,PCHMG=-1 $
   COND    LBL7,PCHMG $
   OUTPUT3 MGG,,,,//C,N,0/C,Y,N1=MMG $
   LABEL   LBL7 $
   PARAM //C,N,NOP/V,Y,PCHMD=-1 $
   COND    LBL8,PCHMD $
   OUTPUT3 PHIG,,,,//C,N,0/C,Y,N1=PHG $
   LABEL   LBL8 $
   PARAM //C,N,NOP/V,Y,PCHMM=-1 $
   COND    LBL9,PCHMM $
   OUTPUT3 KHH,BHH,MHH,,//C,N,0/C,Y,N1=KKH/C,Y,N2=BBH/C,Y,N3=MMH $
   LABEL   LBL9 $
   JUMP    FINIS$
   LABEL   ERRO2 $
   PRTPARM //C,N,-2/C,N,MODES $
   LABEL   FINIS $
   END     $
CEND
```

Because NASTRAN is extremely inflexible in permitting the user to control output data formatting, a preprocessor program must be available for reading the NASTRAN-generated data. The next program is designed to not only read the data but also to process the data and compute the input data required by both N-BOD2 and DISCOS in respectively acceptable format.

For DISCOS users, the mode-dependent parameters are outputted on the line printer and on magnetic tape (data set 2) in a DISCOS-compatible format.

```
C
C
C     DISCOS, N-BOD2 PREPROCESSOR OF NASTRAN GENERATED DATA
C
C        PURPOSE:
C                1) READ ONE OR MORE NASTRAN GENERATED TAPES
C                2) WRITE ONE INPUT TAPE FOR DISCOS PROGRAM
C                     CONTAINING FLEXIBLE BODY DATA FOR ALL
C                     FLEXIBLE BODIES IN SIMULATION
C                3) COMPUTE RESULTANT MODE DEPENDENT PARAMETERS
C                     FOR GSFC MULTI-FLEXIBLE BODY PROGRAM
C                     N-BOD2, THIS SECTION OF NO INTEREST TO
C                     DISCOS USERS
C
C
C        DEFINITIONS:
C
C     DATA SET  1 = SCRATCH FILE USED TO PROCESS NASTRAN DATA
C
C     DATA SET  2 = DISCOS INPUT TAPE (THE OUTPUT TAPE OF THIS PROGRAM)
C        THE CARD OUTPUT OF THIS PROGRAM ASSUMES THAT DATA SET 14 IN
C        DISCOS WILL BE RESERVED FOR THE FLEXIBLE BODY DATA
C
C     DATA SET  3 = SCRATCH FILE USED TO REDUCE CORE REQUIREMENTS
C
C     DATA SET 11 = LOCATION IN SYSTEM OF NASTRAN TAPE ASSOCIATED WITH
C                   THE FLEXIBLE BODY IN DISCOS HAVING LOWEST MAGNITUDE
C                   INTEGER LABEL
C
C     DATA SET 12 = LOCATION IN SYSTEM OF NASTRAN TAPE ASSOCIATED WITH
C                   THE NEXT FLEXIBLE BODY IN THE DISCOS MODEL
C            .
C            .
C            .
C          ETC.     PROGRAM DIMENSIONED FOR A MAXIMUM OF 6 NASTRAN TAPES
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DATA IDL/1H$/
      DATA ISTR,IBLK/1H*,1H /
      DATA  IGPL,IBGP,IMGG     /4HGPL ,4HBGPD,4HMGG /
      DATA ID,IT,II/1HD,1HT,1HI/
      DATA IR,IE,IS/1HR,1HE,1HS/
C
      INTEGER  INPT(80), IGRID(500), IA(20), IB(20),
     *         ICOT(5),  IY(4)
      INTEGER MSAVE(100),MSTOTL,HINGE
C
      REAL*8   X(500),    Y(500),     Z(500),
     *         INERTA(6,6,500),       PHI(6,500,12),
     *         KHH(12,12),            BHH(12,12),
     *         MHH(12,12),            XX(4),          XY(4)
C
      REAL*8 AMD(500,72)
C
      COMMON /WORK/  AMD
C
      EQUIVALENCE (AMD(1,1),PHI(1,1,1))
      EQUIVALENCE (AMD(1,1),X(1))
      EQUIVALENCE (AMD(1,2),Y(1))
      EQUIVALENCE (AMD(1,3),Z(1))
      EQUIVALENCE (AMD(1,4),INERTA(1,1,1))
C
      LOGICAL  LECHO,LPUNCH,LCHECK.LNAST
      LOGICAL LRSTRT
C
C
  100 FORMAT (I5,3L5,2I5,L5)
  101 FORMAT (4A4,7I8,2A4)
  102 FORMAT (2A4,8I8)
  103 FORMAT (4A4,5I8,E16.9)
  104 FORMAT (2A4,2E16.9,I16,E16.9)
  105 FORMAT (2A4,2E16.9)
  106 FORMAT (I10)
  108 FORMAT (4A4,I8,2I16,E16.8)
  109 FORMAT (10E13.5)
  110 FORMAT (6E15.5)
  111 FORMAT (12I5)
  112 FORMAT ('1',I5,3L5,2I5,L5)
  113 FORMAT (1X,20I5)
  114 FORMAT ('1',5X,'******  MODAL DATA SELECTED FROM NASTRAN INPUT TAP
     *E',I4,' FOR PROCESSING  ******',//)
  115 FORMAT (12X,'MODE',I4,' LISTED BELOW IS MODE',I4,' ON THE NASTRAN
     *DATA TAPE')
  200 FORMAT (' NJOINT =',I5,' NUMBER OF GRID POINT LOCATIONS TO BE READ
     * =',I5,' THESE SHOULD BE EQUAL')
  201 FORMAT ('     NUMBER    GRID POINT    X-LOCATION     Y-LOCATION
     * Z-LOCATION ')
  202 FORMAT (6X,I5,7X,I5,2X,3E15.5)
  300 FORMAT (80A1)
  302 FORMAT (4A4,I8,2I16,E16.8,2A4)
  303 FORMAT (2A4,4E16.8,2A4)
  400 FORMAT (2X,80A1)
  500 FORMAT (' ANOMOLOUS EOF MARK ON TAPE, LISTING FOLLOWS ')
  501 FORMAT (/,' INTERNAL GRID POINT ',I5,' EXTERNAL GRID POINT',I5,' I
     *N THE NASTRAN BULK DATA DECK')
  502 FORMAT (' X,Y,Z COORDINATES')
  503 FORMAT (6E20.10)
  504 FORMAT (' LUMPED INERTIA MATRIX')
  505 FORMAT (' PHI (6 MODAL COMPONENTS FOR EACH SELECTED MODE)')
  506 FORMAT (//,' KHH ')
  507 FORMAT ('1',5X,'****** LUMPED PARAMETER AND MODAL DATA TO BE PROCE
     *SSED FOR DISCOS AND N-BOD2 INPUT DATA ******',///)
  508 FORMAT (//,' BHH ')
  509 FORMAT (//,' MHH ')
  510 FORMAT (/////)
  511 FORMAT (2X,'EXIT DECODE ',4(I8,E16.8))
  512 FORMAT (2X,'ENTER DECODE',2I8,4E16.8)
  600 FORMAT (I5,3E20.8)
C
C
C                   READ TWO INPUT CARDS PER NASTRAN TAPE
C                     NTAPE = DATA SET WHERE TAPE IS TO
C                              BE FOUND NTAPE=11,12,....
C                     LECHO = .TRUE. PRINT ECHO OF PROCESSED DATA
C                            = .FALSE. PRINT ONLY IF ERROR FOUND ON TAPE
C                     LPUNCH = .TRUE. PUNCH PLOT DATA
C                             = .FALSE. DON'T
C                     LCHECK = .TRUE. PRINT DATA IN/OUT OF DECODE
C                             = .FALSE. DON'T
C                     HINGE = GRID POINT NUMBER OF HINGE POINT AS
C                              DEFINED BY USER IN NASTRAN BULK DATA
C                     MSTOTL = TOTAL NUMBER OF FLEXIBLE BODY MODES TO
C                               PROCESS FOR DISCOS AND N-BOD2 DATA
C                     LNAST = .TRUE. PRINT NASTRAN DATA INPUT
C                              .FALSE. DON'T
C                     MSAVE  = MODE SAVE ARRAY
C                                MSAVE(K) = 0 SKIP MODE K DATA
C                                           K PROCESS MODE K DATA
C
      REWIND 2
    1 CONTINUE
      READ(5,100,END=4) NTAPE,LECHO,LPUNCH,LCHECK,HINGE,MSTOTL,LNAST
      WRITE(6,112)      NTAPE,LECHO,LPUNCH,LCHECK,HINGE,MSTOTL,LNAST
      READ(5,111) (IA(I),I=1,MSTOTL)
      DO 11 I=1,100
   11 MSAVE(I) = 0
      DO 12 I=1,MSTOTL
   12 MSAVE(IA(I)) = 1
      WRITE(6,113) MSAVE
      REWIND NTAPE
C                        LRSTRT = .FALSE.  NO RESTART CARDS ON NTAPE
C                               = .TRUE.    RSTART CARDS ON NTAPE
      ICNTRS = 0
      LRSTRT = .FALSE.
    9 READ (NTAPE,300,END=8) INPT
      IF(LNAST) WRITE(6.400) INPT
C     CHECK FOR RESTART CARDS
      IF(IR.NE.INPT(1)) GO TO 9
      IF(IE.NE.INPT(2)) GO TO 9
      IF(IS.NE.INPT(3)) GO TO 9
      IF(IT.NE.INPT(4)) GO TO 9
C        COME HERE FOR RESTART CARDS. COUNT THEM AND SET FLAG
      LRSTRT = .TRUE.
      ICNTRS = ICNTRS+1
   13 READ (NTAPE,300,END=8) INPT
      IF(LNAST) WRITE(6.400) INPT
      IF(ID.EQ.INPT(1).AND.IT.EQ.INPT(2).AND.II.EQ.INPT(3)) GO TO 9
      ICNTRS =ICNTRS+1
      GO TO 13
    8 CONTINUE
      REWIND NTAPE
      REWIND 1
      REWIND 3
      ICNT = 0
      IBLK = 0
      DO 7 I=1,5
    7 ICOT(I) = 0
C
C                   READ ALL DTI CARDS FROM NASTRAN TAPE NTAPE TO OBTAIN
C                        1) GRID POINT NUMBERING SEQUENCE SET UP BY NASTRAN
C                        2) GRID POINT LOCATIONS
C
C              **** NOTE ****
C
C                   NASTRAN WILL RENUMBER GRID POINTS DEFINED BY USER
C                   INTO A SEQUENCIAL SET, THIS IS USED FOR INTERNAL NASTRAN
C                   COMPUTATION AND WILL BE USED BY DISCOS
C
C     SKIP RESTART CARDS IN FRONT OF GOOD DATA
      IF(.NOT.LRSTRT) GO TO 2
      DO 14 I=1,ICNTRS
   14 READ (NTAPE,300,END=3) INPT
    2 READ (NTAPE,101,END=3) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
C          CHECK FOR TABLE HEADING
      IF(IA(3).EQ.IGPL) GO TO 50
      IF(IA(3).EQ.IBGP) GO TO 60
      IF(IA(3).EQ.IMGG) GO TO 70
      GO TO 2
C
C      ERROR ON TAPE
    3 CONTINUE
      WRITE(6,500)
    6 REWIND NTAPE
    5 READ (NTAPE,300,END=4) INPT
      WRITE(6,400) INPT
      GO TO 5
    4 STOP
C
C                     COME HERE TO PROCESS GPL TABLE DATA
   50 CONTINUE
      ICNT = ICNT + 1
      IF(ICNT.NE.1) GO TO 51
      NJOINT = IB(2)
C                                NJOINT = NUMBER OF GRID POINTS IN MODEL
      GO TO 2
   51 IF(ICNT.NE.2)GO TO 52
      DO 53 J=1,6
   53 IGRID(J) = IB(J+1)
      NC = (NJOINT-6)/8
      DO 54 J=1,NC
      JJ = 7 + 8*(J-1)
      JJ7 = JJ+7
   54 READ(NTAPE,102) (IA(I),I=1,2),(IGRID(I),I=JJ,JJ7)
      JJ = 6 + 8*NC
      JJ1 = JJ+1
      IF(JJ.EQ.NJOINT) GO TO 2
      READ(NTAPE,102) (IA(I),I=1,2),(IGRID(I),I=JJ1,NJOINT)
      GO TO 2
C                                IGRID(I) =USER DEFINED GRID POINT NUMBER
```

```
C                                         WHICH NASTRAN HAS RELABLED TO
C                                         BE INTERNAL NUMBER I
   52 ICNT = 0
      NC = (NJOINT-3)/4
      IF(NJOINT.EQ.4*NC+3) GO TO 16
      NC = NC + 1
   16 DO 15 I=1,NC
   15 READ (NTAPE,300) INPT
      GO TO 2
C
C
C                COME HERE TO PROCESS BGPDT CARDS
   60 CONTINUE
      IF(IB(2).EQ.NJOINT) GO TO 61
C     ERROR
      WRITE(6,200) NJOINT,IB(2)
      GO TO 6
   61 READ (NTAPE,300) INPT
      READ (NTAPE,103) (IA(I),I=1,4),(IB(I),I=1,5),X(1)
      DO 62 J=2,NJOINT
   62 READ (NTAPE,104) (IA(I),I=1,2),Y(J-1),Z(J-1),IB(1),X(J)
      J = NJOINT
      READ (NTAPE,105) (IA(I),I=1,2),Y(J),Z(J)
      GO TO 2
C        SAVE LOCATION DATA, WRITE IT ON TAPE 3 AFTER INERTIA DATA
C
C               X(J) = X-COORDINATE OF GRID POINT IGRID(J)
C               Y(J) = Y-COORDINATE OF GRID POINT IGRID(J)
C               Z(J) = Z-COORDINATE OF GRID POINT IGRID(J)
C
C
C                   COME HERE TO PROCESS ALL DMI CARDS
C        STEP 1:
C                COUNT CARDS IN EACH GROUP
C                ASCERTAIN FORMAT OF EACH CARD
C                REWRITE ON SCRATCH FILE ALL CARDS
C                 PRECEEDED BY THEIR FORMAT CODE
C        STEP 2:
C                REWIND SCRATCH FILE
C                READ CARDS OFF SCRATCH FILE WITH KNOWN FORMAT
C                LOAD ALL ARRAYS, SKIPPING ALL UNWANTED MODES
   70 CONTINUE
      REWIND 1
      IFORM = 301
      WRITE(1,106) IFORM
      WRITE(1,101) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
      IBLK = 1
      ICOT(IBLK) = 1
      IF(IB(2).NE.6.OR.IB(6).NE.IB(7).OR.IB(6).NE.NJOINT*6) GO TO 71
      DO 76 I=1,NJOINT
      DO 76 J1=1,6
      DO 76 J2=1,6
   76 INERTA(J1,J2,I) =0.0
      GO TO 72
   71 CONTINUE
C     ERROR
      WRITE(6,101) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
      GO TO 6
   72 READ(NTAPE,300,END=80) INPT
      IF(INPT(4).NE.ISTR) GO TO 73
      ICOT(IBLK) = ICOT(IBLK) + 1
      IFORM = 302
      GO TO 74
   73 IF(INPT(1).EQ.ISTR) GO TO 75
C              END OF DATA BLOCK IBLK
C        CHECK FOR 'END OF CHECKPOINT DICTIONARY'
      IF(INPT(1).EQ.IDL) GO TO 72
      IBLK = IBLK + 1
      ICOT(IBLK) = ICOT(IBLK) + 1
      IFORM = 301
      GO TO 74
   75 CALL INTERP(INPT,IFORM)
      ICOT(IBLK) = ICOT(IBLK) + 1
   74 WRITE(1,106) IFORM
      WRITE(1,300) INPT
      GO TO 72
   80 END FILE 1
C
C                  ALL DMI CARDS READ, FORMAT DEDUCED AND
C                  REWRITTEN ON SCRATCH FILE 1
C     FORMAT CODES:
C           301 = FORMAT (4A4,7I8,2A4)
C           302 = FORMAT (4A4,I8,2I16,E16.8,2A4)
C           303 = FORMAT (2A4,4E16.8,2A4)
C           304 = FORMAT (2A4,3E16.8,I16,2A4)
C           305 = FORMAT (2A4,2E16.8,I16,E16.8,2A4)
C           306 = FORMAT (2A4,E16.8,I16,2E16.8,2A4)
C           307 = FORMAT (2A4,E16.8,I16,E16.8,I16,2A4)
C           308 = FORMAT (2A4,I16.8,3E16.8,2A4)
C           309 = FORMAT (2A4,I16.8,2E16.8,I16.8,2A4)
C           310 = FORMAT (2A4,I16,E16.8,I16,E16.8,2A4)
C
C
C                       START TO PROCESS DMI CARDS
      REWIND 1
      IF(IBLK.LE.5) GO TO 20
C        ERROR IBLK MUST BE LESS THAN OR EQUAL TO 5
   81 REWIND 1
   83 READ (1,300,END=90) INPT
      WRITE(6,400) INPT
      GO TO 83
   90 STOP
   20 CONTINUE
      DO 22 IBK=1,IBLK
      ICBLK = ICOT(IBK)
      DO 21 JJ =1,ICBLK
      IF(JJ.NE.1) GO TO 23
C          READ FIRST CARD OF DATA BLOCK
      READ (1,106) IFORM
      READ (1,101) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
      IF(IBK.NE.1) GO TO 40
C
C                 INERTIA MATRIX
      IF(IB(2).NE.6.OR.IB(6).NE.IB(7).OR.IB(6).NE.NJOINT*6) GO TO 81
C          SO FAR SO GOOD
      GO TO 21
   40 IF(IBK.NE.2) GO TO 41
C
C                 MODE SHAPES
      IF(IB(6).NE.6*NJOINT) GO TO 81
      NMODES = IB(7)
C                             NMODES = NUMBER OF MODES
C                             MSTOTL = NUMBER OF MODES TO PROCESS
      DO 42 I1=1,6
      DO 42 I2=1,NJOINT
      DO 42 I3=1,MSTOTL
   42 PHI(I1,I2,I3) = 0.0
      GO TO 21
C
C             MODAL MASS, STIFFNESS AND DAMPING MATRICES
   41 IF(IB(6).NE.IB(7).OR.IB(7).NE.NMODES) GO TO 71
      DO 43 I1=1,MSTOTL
      DO 43 I2=1,MSTOTL
      IF(IBK.NE.3) GO TO 44
      KHH(I1,I2) = 0.0
      GO TO 43
   44 IF(IBK.NE.4) GO TO 45
      BHH(I1,I2) = 0.0
      GO TO 43
   45 MHH(I1,I2) = 0.0
   43 CONTINUE
      GO TO 21
C
C                  READ IN DATA CARDS
   23 CONTINUE
      READ (1,106) IFORM
      IF(IFORM.NE.302) GO TO 24
      READ (1,302) (IA(I),I=1,4),IB(1),NNR,NNC,XX(1),(IA(I),I=5,6)
      NSR = MSAVE(NNR)
      NSC = MSAVE(NNC)
      IF(IBK.NE.1) GO TO 25
      N1T = (NNC-1)/6 + 1
      N2T = (NNR-1)/6 + 1
      IF(N1T.EQ.N2T) GO TO 26
C      ERROR
      WRITE(6,302) (IA(I),I=1,4),IB(1),NNR,NNC,XX(1),(IA(I),I=5,6)
      GO TO 81
   26 NT = N1T
      NC = MOD(NNC-1,6) + 1
      NR = MOD(NNR-1,6) + 1
      INERTA(NR,NC,NT) = XX(1)
      GO TO 21
   25 IF(IBK.NE.2) GO TO 27
      NT = (NNC-1)/6 + 1
      NE = MOD(NNC-1,6) + 1
      IF(NSR.EQ.0) GO TO 21
      PHI(NE,NT,NSR) = XX(1)
      GO TO 21
   27 IF(IBK.NE.3) GO TO 28
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 21
      KHH(NSR,NSC) = XX(1)
      GO TO 21
   28 IF(IBK.NE.4) GO TO 29
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 21
      BHH(NSR,NSC) = XX(1)
      GO TO 21
   29 CONTINUE
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 21
      MHH(NSR,NSC) = XX(1)
      GO TO 21
```

```
C
   24 CONTINUE
C         FORMAT 4E16.8 FOR XX WILL READ INTEGER I WHICH IS RIGHT
C         JUSTIFIED AS I * 1.0E-08
C         THIS IS EASILY UNDONE IN DECODE
      READ (1,303) (IA(I),I=1,2),(XX(I),I=1,4),(IA(I),I=3,4)
      IF(LCHECK) WRITE (6,512) IFORM,NNC,(XX(I),I=1,4)
      CALL DECODE(IFORM,XX,NNC,XY,IY,NY)
      IF(LCHECK) WRITE (6,511) (IY(I),XY(I),I=1,NY)
      DO 30 I=1,NY
      IF(IBK.NE.1) GO TO 31
      NC = MOD(IY(I)-1,6) + 1
      INERTA(NR,NC.NT) = XY(I)
      GO TO 30
   31 IF(IBK.NE.2) GO TO 32
      NT = (IY(I)-1)/6 + 1
      NE = MOD(IY(I)-1,6) + 1
      IF(NSR.EQ.0)  GO TO 30
      PHI(NE,NT,NSR) = XY(I)
      GO TO 30
   32 IF(IBK.NE.3) GO TO 33
      NSC = MSAVE(IY(I))
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 30
      KHH(NSR,NSC) = XY(I)
      GO TO 30
   33 IF(IBK.NE.4) GO TO 34
      NSC = MSAVE(IY(I))
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 30
      BHH(NSR,NSC) = XY(I)
      GO TO 30
   34 CONTINUE
      NSC = MSAVE(IY(I))
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 30
      MHH(NSR,NSC) = XY(I)
   30 CONTINUE
   21 CONTINUE
C
C
      IF(IBK.NE.1) GO TO 63
      REWIND 3
      WRITE (3) (((INERTA(I,J,N),I=1,6),J=1,6),N=1,NJOINT)
      WRITE (3) (X(N),Y(N),Z(N),N=1,NJOINT)
      IF(.NOT.LPUNCH) GO TO 10
      WRITE (7,600) (IGRID(N),X(N),Y(N),Z(N),N=1,NJOINT)
   10 CONTINUE
      WRITE(6,507)
      DO 64 N=1,NJOINT
      WRITE (6,501) N,IGRID(N)
      WRITE (6,502)
      WRITE (6,503) X(N),Y(N),Z(N)
      WRITE (6,504)
      WRITE (6,503) ((INERTA(I,J,N),J=1,6),I=1,6)
   64 CONTINUE
      WRITE (6,510)
      GO TO 22
   63 IF(IBK.NE.2) GO TO 65
      WRITE (3) (((PHI(I,N,J),I=1,6),J=1,MSTOTL),N=1,NJOINT)
      IF(.NOT.LPUNCH) GO TO 67
      DO 68 J=1,MSTOTL
      WRITE (7,600) (IGRID(N),(PHI(I,N,J),I=1,3),N=1,NJOINT)
   68 CONTINUE
   67 CONTINUE
      WRITE (6,114) NTAPE
      DO 17 I=1,NMODES
      IF(MSAVE(I).EQ.0) GO TO 17
      WRITE (6,115) MSAVE(I),I
   17 CONTINUE
      WRITE (6,510)
      DO 66 N=1,NJOINT
      WRITE (6,501) N,IGRID(N)
      WRITE (6,505)
      WRITE (6,503) ((PHI(I,N,J),I=1,6),J=1,MSTOTL)
   66 CONTINUE
      WRITE (6,510)
      GO TO 22
   65 IF(IBK.NE.3) GO TO 69
      WRITE (3) ((KHH(I,J),I=1,MSTOTL),J=1,MSTOTL)
      WRITE (6,506)
      WRITE (6,503) ((KHH(I,J),J=1,MSTOTL),I=1,MSTOTL)
      WRITE (6,510)
      GO TO 22
   69 IF(IBK.NE.4) GO TO 77
      WRITE (3) ((BHH(I,J),I=1,MSTOTL),J=1,MSTOTL)
      WRITE (6,508)
      WRITE (6,503) ((BHH(I,J),J=1,MSTOTL),I=1,MSTOTL)
      WRITE (6,510)
      GO TO 22
   77 IF(IBK.NE.5) GO TO 22
      WRITE (3) ((MHH(I,J),I=1,MSTOTL),J=1,MSTOTL)
      WRITE (6,509)
      WRITE (6,503) ((MHH(I,J),J=1,MSTOTL),I=1,MSTOTL)
      WRITE (6,510)
   22 CONTINUE
C
C                        DATA READY FOR DISCOS PROCESSING
      CALL DISCOS(IGRID,NJOINT,MSTOTL,ITAPE,LECHO)
      CALL NBOD2 (IGRID,NJOINT,MSTOTL,ITAPE,LECHO,HINGE)
      IF(ITAPE.LE.6) GO TO 1
      STOP
      END




      SUBROUTINE DISCOS(IGRID,NJ,NM,ICNT,LECHO)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      LOGICAL LECHO
C
      REAL*8   X(500),Y(500),Z(500),INERTA(6,6,500),IRUNNO
      REAL*8   PHI(6,500,12),KHH(12,12),BHH(12,12),LOC(6)
      REAL*8   AMS(6),JMASS(500,1),JINER(500,6),AIN(6),SMM(6)
      REAL*8   SMSM(500,3),LOCA(500,3),MOD(6,6),AMD(500,13),KH(6),BH(6)
      REAL*8 AMP(500,72)
      COMMON /WORK/  AMP
      EQUIVALENCE (AMP(1,1),X(1))
      EQUIVALENCE (AMP(1,2),Y(1))
      EQUIVALENCE (AMP(1,3),Z(1))
      EQUIVALENCE (AMP(1,4),INERTA(1,1,1))
      EQUIVALENCE (AMP(1,1),PHI(1,1,1))
C
      INTEGER   IGRID(500)
C
      COMMON /WORK2/ AMD
      EQUIVALENCE (AMD(1,1),JMASS(1,1)),     (AMD(1,1),JINER(1,1))
      EQUIVALENCE (AMD(1,7),SMSM(1,1)),      (AMD(1,10),LOCA(1,1))
      EQUIVALENCE (AMD(1,13),KHH(1,1)),  (AMD(145,13),BHH(1,1))
C
      DATA       IFST/0/
      DATA       AMS/6HJMASS1,6HJMASS2,6HJMASS3,6HJMASS4,6HJMASS5,
     *                6HJMASS6/
      DATA      FLX/6HFLXDAT/
      DATA      AIN/6HINERT1,6HINERT2,6HINERT3,6HINERT4,6HINERT5,
     *                6HINERT6/
      DATA       SMM/6HSTMSM1,6HSTMSM2,6HSTMSM3,6HSTSMS4,6HSTMSM5,
     *                6HSTMSM6/
      DATA      LOC/6HLOCAT1,6HLOCAT2,6HLOCAT3,6HLOCAT4,6HLOCAT5,
     *                6HLOCAT6/
      DATA   MOD/6HXDIS1 ,6HYDIS1 ,6HZDIS1 ,6HTHA11 ,6HTHA21 ,6HTHA31 ,
     *              6HXDIS2 ,6HYDIS2 ,6HZDIS2 ,6HTHA12 ,6HTHA22 ,6HTHA32 ,
     *              6HXDIS3 ,6HYDIS3 ,6HZDIS3 ,6HTHA13 ,6HTHA23 ,6HTHA33 ,
     *              6HXDIS4 ,6HYDIS4 ,6HZDIS4 ,6HTHA14 ,6HTHA24 ,6HTHA34 ,
     *              6HXDIS5 ,6HYDIS5 ,6HZDIS5 ,6HTHA15 ,6HTHA25 ,6HTHA35 ,
     *              6HXDIS6 ,6HYDIS6 ,6HZDIS6 ,6HTHA16 ,6HTHA26 ,6HTHA36 /
      DATA    KH/6HKHH1   ,6HKHH2   ,6HKHH3   ,6HKHH4   ,6HKHH5   ,6HKHH6   /
      DATA    BH/6HBHH1   ,6HBHH2   ,6HBHH3   ,6HBHH4   ,6HBHH5   ,6HBHH6   /
      DATA IRUNNO/6HPREPRS/
C
  700 FORMAT (A6,'    0    14PREPRS ')
C
C                  INITIALIZE TAPE, START COUNTER
      REWIND 3
      IF(IFST.NE.0) GO TO 1
      CALL INTAPE(2,FLX)
      ICNT = 0
      IFST = 1
    1 CONTINUE
      ICNT = ICNT+1
C
C                       PROCESS MASS MATRIX DATA
      READ   (3) (((INERTA(I,J,N),I=1,6),J=1,6),N=1,NJ)
      READ   (3) (X(N),Y(N),Z(N),N=1,NJ)
      DO 2  I=1,NJ
      IF(INERTA(1,1,I).NE.INERTA(2,2,I)) GO TO 3
      IF(INERTA(2,2,I).NE.INERTA(3,3,I)) GO TO 3
      IF(INERTA(3,3,I).LT.0) GO TO 3
      JMASS(I,1) = INERTA(3,3,I)
      GO TO 2
    3 WRITE(6,500) I,(INERTA(II,II,I),II=1,3)
  500 FORMAT (' ERROR MASS MATRIX POINT ',I5,' JMASS =',3E20.10)
      JMASS(I,1) = (INERTA(1,1,I)+INERTA(2,2,I)+INERTA(3,3,I))/3.0D0
    2 CONTINUE
C                       PUNCH READ MATRIX INPUT CARD
C                       PUT MASS MATRIX ON TAPE
      WRITE(7,700) AMS(ICNT)
      IF(LECHO) CALL WRITE(JMASS,NJ,1,AMS(ICNT),500)
      CALL WTAPE(JMASS,NJ,1,AMS(ICNT),500,2)
C
```

```
C                         PROCESS INERTIA DYAD DATA
      DO 4  I=1,NJ
      JINER(I,1) =  INERTA(4,4,I)
      JINER(I,2) =  INERTA(5,5,I)
      JINER(I,3) =  INERTA(6,6,I)
      JINER(I,4) =-INERTA(4,5,I)
      JINER(I,5) =-INERTA(4,6,I)
    4 JINER(I,6) =-INERTA(5,6,I)
      WRITE(7,700) AIN(ICNT)
      IF(LECHO) CALL WRITE(JINER,NJ,6,AIN(ICNT),500)
      CALL WTAPE(JINER,NJ,6,AIN(ICNT),500,2)
C
C                         PROCESS STATIC MASS MOMENT DATA
C                         PROCESS JOINT LOCATION DATA
      DO 5  I=1,NJ
      SMSM(I,1) =   INERTA(2,6,I)
      SMSM(I,2) = -INERTA(1,6,I)
      SMSM(I,3) =   INERTA(1,5,I)
      LOCA(I,1) =   X(I)
      LOCA(I,2) =   Y(I)
    5 LOCA(I,3) =   Z(I)
      WRITE(7,700)  SMM(ICNT)
      IF(LECHO) CALL WRITE(SMSM ,NJ,3,SMM(ICNT),500)
      CALL WTAPE(SMSM,NJ,3,SMM(ICNT),500,2)
      WRITE(7,700)  LOC(ICNT)
      IF(LECHO) CALL WRITE(LOCA ,NJ,3,LOC(ICNT),500)
      CALL WTAPE(LOCA,NJ,3,LOC(ICNT),500,2)
C
C                         PROCESS MODAL DATA
      READ  (3) (((PHI(I,N,J),I=1,6),J=1,NM),N=1,NJ)
      DO 6 K=1,6
      DO 7 I=1,NJ
      DO 7 M=1,NM
    7 AMD(I,M) = PHI(K,I,M)
      WRITE(7,700)  MOD(K,ICNT)
      IF(LECHO) CALL WRITE(AMD,NJ,NM,MOD(K,ICNT),500)
      CALL WTAPE(AMD,NJ,NM,MOD(K,ICNT),500,2)
    6 CONTINUE
C
C                         MODAL MATRICES
      READ  (3) ((KHH(I,J),I=1,NM),J=1,NM)
      READ  (3) ((BHH(I,J),I=1,NM),J=1,NM)
      WRITE(7,700)  KH(ICNT)
      IF(LECHO) CALL WRITE(KHH,NM,NM,KH(ICNT),12)
      CALL WTAPE(KHH,NM,NM,KH(ICNT),12,2)
      WRITE(7,700)  BH(ICNT)
      WRITE(6,100)
      IF(LECHO) CALL WRITE(BHH,NM,NM,BH(ICNT),12)
      CALL WTAPE(BHH,NM,NM,BH(ICNT),12,2)
C
      CALL LTAPE(2)
      CALL RTAPE(IRUNNO,AMS(ICNT),AMD,NJ,1,500,12,2)
      CALL WRITE(AMD,NJ,1,AMS(ICNT),500)
      CALL RTAPE(IRUNNO,AIN(ICNT),AMD,NJ,6,500,12,2)
      CALL WRITE(AMD,NJ,6,AIN(ICNT),500)
      CALL RTAPE(IRUNNO,SMM(ICNT),AMD,NJ,3,500,12,2)
      CALL WRITE(AMD,NJ,3,SMM(ICNT),500)
      CALL RTAPE(IRUNNO,LOC(ICNT),AMD,NJ,3,500,12,2)
      CALL WRITE(AMD,NJ,3,LOC(ICNT),500)
      DO 8 K=1,6
      CALL RTAPE(IRUNNO,MOD(K,ICNT),AMD,NJ,NM,500,12,2)
    8 CALL WRITE(AMD,NJ,NM,MOD(K,ICNT),500)
      CALL RTAPE(IRUNNO,KH(ICNT),KHH,NM,NM,12,12,2)
      CALL WRITE(KHH,NM,NM,KH(ICNT),12)
      CALL RTAPE(IRUNNO,BH(ICNT),BHH,NM,NM,12,12,2)
      CALL WRITE(BHH,NM,NM,BH(ICNT),12)
  100 FORMAT ('1',5X,'****** PROCESSED LUMPED PARAMETER AND MODAL DATA N
     *OW ON DATA SET 2 (DISCOS INPUT TAPE) ******',///)
      RETURN
      END




      SUBROUTINE NBOD2 (IGRID,NJ,NM,ICNT,LECHO,HINGE)
C     COMPUTE RESULTANT MODE DEPENDENT PARAMETERS FOR N-BOD2
      IMPLICIT REAL*8 (A-H,O-Z)
C
      LOGICAL LECHO
C
      COMMON /WORK/  AMP
      REAL*8 AMP(500,72)
      COMMON /WORK2/ AMD
      REAL*8 AMD(500,13)
C
      REAL*8    AMS(6),FLX,AIN(6),SMM(6),LOC(6),MOD(6,6),KH(6),BH(6)
      REAL*8    JMASS(500), JLOC(3,500), CMLOC(3), JINERT(3,3,500),
     *          PHIT(3,500,12), PHIR(3,500,12),  SMO(3,500),
     *          FLOM(12), ZETA(12), BDINER(3,3),
     *          FLA(3,12), FLB(3,12), FLC(3,12), FLD(3,3,12),
     *          FLJ(3,3,12),
     *          FCF(3,3,12,12), FCK(3,12,12),
     *          TEM(3), TEM1(3,3), ADM(500,12,6), TEM(3),
     *          KHH(12,12), BHH(12,12), IRUNNO
      INTEGER   IGRID(500)
      INTEGER   HINGE, INTH
C
      DATA      AMS/6HJMASS1,6HJMASS2,6HJMASS3,6HJMASS4,6HJMASS5,
     *              6HJMASS6/
      DATA     AIN/6HINERT1,6HINERT2,6HINERT3,6HINERT4,6HINERT5,
     *              6HINERT6/
      DATA      SMM/6HSTMSM1,6HSTMSM2,6HSTMSM3,6HSTSMS4,6HSTMSM5,
     *              6HSTMSM6/
      DATA      LOC/6HLOCAT1,6HLOCAT2,6HLOCAT3,6HLOCAT4,6HLOCAT5,
     *              6HLOCAT6/
      DATA   MOD/6HXDIS1 ,6HYDIS1 ,6HZDIS1 ,6HTHA11 ,6HTHA21 ,6HTHA31 ,
     *            6HXDIS2 ,6HYDIS2 ,6HZDIS2 ,6HTHA12 ,6HTHA22 ,6HTHA32 ,
     *            6HXDIS3 ,6HYDIS3 ,6HZDIS3 ,6HTHA13 ,6HTHA23 ,6HTHA33 ,
     *            6HXDIS4 ,6HYDIS4 ,6HZDIS4 ,6HTHA14 ,6HTHA24 ,6HTHA34 ,
     *            6HXDIS5 ,6HYDIS5 ,6HZDIS5 ,6HTHA15 ,6HTHA25 ,6HTHA35 ,
     *            6HXDIS6 ,6HYDIS6 ,6HZDIS6 ,6HTHA16 ,6HTHA26 ,6HTHA36 /
      DATA    KH/6HKHH1  ,6HKHH2  ,6HKHH3  ,6HKHH4  ,6HKHH5  ,6HKHH6  /
      DATA    BH/6HBHH1  ,6HBHH2  ,6HBHH3  ,6HBHH4  ,6HBHH5  ,6HBHH6  /
      DATA IRUNNO/6HPREPRS/
C
      EQUIVALENCE (AMP(1, 1),PHIT(1,1,1)),(AMP(1,37),PHIR(1,1,1)),
     *              (AMP(1, 1),ADM(1,1,1)),
     *              (AMP(1,13),ADM(1,1,2)),
     *            (AMP(1,25),ADM(1,1,3)),
     *            (AMP(1,37),ADM(1,1,4)),
     *            (AMP(1,49),ADM(1,1,5)),
     *            (AMP(1,61),ADM(1,1,6))
      EQUIVALENCE (AMD(1,1),JMASS(1)), (AMD(1,2),JLOC(1,1)),
     *            (AMD(1,5),SMO(1,1)), (AMD(1,5),JINERT(1,1,1))
C
C        FIND INTERNAL NASTRAN NUMBER ASSOCIATED
C        WITH THE HINGE POINT (GRID POINT 'HINGE')
      DO 1 N=1,NJ
      IF(IGRID(N).EQ.HINGE) GO TO 2
    1 CONTINUE
      WRITE(6,100) HINGE
      RETURN
    2 INTH = N
C
C                 INTH = INTERNAL NASTRAN NUMBER
C                        OF HINGE POINT
C
C         GET DATA OFF OF TAPE 2 TO COMPUTE CENTER OF MASS
C
      CALL RTAPE(IRUNNO,LOC(ICNT),AMP,NJ,3,500,72,2)
C
C            JLOC(I,N) = I-TH COORDINATE OF GRID POINT IGRID(N)
C                        VECTOR RELATIVE TO HINGE POINT HINGE
      DO 3 N=1,NJ
      DO 3 I=1,3
      JLOC(I,N) = AMP(N,I) - AMP(INTH,I)
    3 CONTINUE
C
      CALL RTAPE(IRUNNO,AMS(ICNT),AMP,NJ,1,500,72,2)
C
C            JMASS(N)   = LUMPED MASS AT GRID POINT IGRID(N)
      DO 4 N=1,NJ
      JMASS(N) = AMP(N,1)
    4 CONTINUE
C
      IF(.NOT.LECHO) GO TO 17
      WRITE(6,101)
      WRITE(6,102) (N,IGRID(N),N,(JLOC(I,N),I=1,3),N,JMASS(N),N=1,NJ)
   17 CONTINUE
C
C        COMPUTE CENTER OF MASS LOCATION
      TOTMAS = 0.0
      CMLOC(1) = 0.0
      CMLOC(2) = 0.0
      CMLOC(3) = 0.0
C
C         READ GRID POINT INERTIA TENSORS AND MASS MOMENTS
C
      CALL RTAPE(IRUNNO,SMM(ICNT),AMP,NJ,3,500,72,2)
      DO 11 N=1,NJ
      DO 11 I=1,3
      SMO(I,N) = AMP(N,I)
      IF(SMO(I,N).EQ.0.0) GO TO 11
      WRITE(6,107) (N,(SMO(I,N),I=1,3))
   11 CONTINUE
      IF(LECHO) CALL WRITE(SMO,3,NJ,6H SMO   ,3)
C
      CALL RTAPE(IRUNNO,AIN(ICNT),AMP,NJ,6,500,72,2)
      DO 10 N=1,NJ
      JINERT(1,1,N) =  AMP(N,1)
      JINERT(1,2,N) = -AMP(N,4)
      JINERT(1,3,N) = -AMP(N,5)
      JINERT(2,1,N) = -AMP(N,4)
      JINERT(2,2,N) =  AMP(N,2)
      JINERT(2,3,N) = -AMP(N,6)
      JINERT(3,1,N) = -AMP(N,5)
      JINERT(3,2,N) = -AMP(N,6)
      JINERT(3,3,N) =  AMP(N,3)
   10 CONTINUE
      IF(LECHO) CALL WRITE(JINERT,9,NJ,6HJINERT,9)
C
C
C
C     COMPUTE UNDEFORMED BODY INERTIA TENSOR
C       RELATIVE TO BODY CENTER OF MASS
C
      DO 20 I=1,3
      DO 20 J=1,3
   20 BDINER(I,J) = 0.0
      DO 19 N=1,NJ
      CALL VECSUB(JLOC(1,N),CMLOC,TEM)
      CALL SUEOP(TEM,TEM,JMASS(N),TEM1)
      CALL DYADD(BDINER,TEM1,BDINER)
      CALL DYADD(BDINER,JINERT(1,1,N),BDINER)
   19 CONTINUE
C
C
C         READ IN MODE SHAPES AND RESORT THEM
C
      DO 6 I=1,6
      CALL RTAPE(IRUNNO,MOD(I,ICNT),ADM(1,1,I),NJ,NM,500,12,2)
    6 CONTINUE
      REWIND 3
      WRITE (3) (((ADM(N,M,I),I=1,3),N=1,NJ),M=1,NM)
      WRITE (3) (((ADM(N,M,I),I=4,6),N=1,NJ),M=1,NM)
      REWIND 3
      READ  (3) (((PHIT(I,N,M),I=1,3),N=1,NJ),M=1,NM)
      READ  (3) (((PHIR(I,N,M),I=1,3),N=1,NJ),M=1,NM)
      REWIND 3
C
      IF(.NOT.LECHO) GO TO 18
      WRITE(6,104)
      DO 7 M=1,NM
      WRITE(6,105)
      WRITE(6,106)    (N,M,(PHIT(I,N,M),I=1,3),
     *                 N,M,(PHIR(I,N,M),I=1,3),N=1,NJ)
    7 CONTINUE
   18 CONTINUE
C
      WRITE(6,118)
      WRITE(6,103) TOTMAS
      WRITE(6,115)
      WRITE(6,119) (BDINER(1,J),J=1,3)
      WRITE(6,120) (BDINER(2,J),J=1,3)
      WRITE(6,121) (BDINER(3,J),J=1,3)
      WRITE(6,115)
      WRITE(6,122) (CMLOC(I),I=1,3)
      WRITE(6,115)
      WRITE (6,125) INTH, HINGE
      WRITE (6,115)
C
C          COMPUTE RESULTANT PARAMETERS
C
      DO 8 M=1,NM
      DO 8 I=1,3
      FLA(I,M) = 0.0
      FLB(I,M) = 0.0
      FLC(I,M) = 0.0
      DO 8 J=1,3
      FLJ(I,J,M) = 0.0
      FLD(I,J,M) = 0.0
    8 CONTINUE
      DO 9 M=1,NM
      DO 9 MM=1,NM
      DO 9 I=1,3
      FCK(I,M,MM) = 0.0
      DO 9 J=1,3
      FCF(I,J,M,MM) = 0.0
    9 CONTINUE
C
C
      AMASS = 1.0/TOTMAS
C
      DO 12 M=1,NM
      DO 13 J=1,NJ
C
      CALL SCLV(JMASS(J),PHIT(1,J,M),TEM)
      CALL VECADD(FLA(1,M),TEM,FLA(1,M))
C
      CALL VECSUB(JLOC(1,J),CMLOC,TEM)
      CALL SUEOP(PHIT(1,J,M),TEM,JMASS(J),TEM1)
      CALL DYADD(FLD(1,1,M),TEM1,FLD(1,1,M))
C
      CALL VECROS(JLOC(1,J),PHIT(1,J,M),TEM)
      CALL SCLV(JMASS(J),TEM,TEM)
      CALL VECADD(FLB(1,M),TEM,FLB(1,M))
C
      IF(JINERT(1,1,J).EQ.0.0.AND.
     *   JINERT(2,2,J).EQ.0.0.AND.
     *   JINERT(3,3,J).EQ.0.0) GO TO 13
C
      CALL DYDOTV(JINERT(1,1,J),PHIR(1,J,M),TEM)
      CALL VECADD(FLC(1,M),TEM,FLC(1,M))
C
      CALL VECXDY(PHIR(1,J,M),JINERT(1,1,J),TEM1)
      CALL DYADD(FLJ(1,1,M),TEM1,FLJ(1,1,M))
   13 CONTINUE
C
C          RESULTANT UNCOUPLED MODE DEPENDED PARAMETERS
C
```

```
      CALL SCLV(AMASS,FLA(1,M),FLA(1,M))
      CALL SCLV(AMASS,FLB(1,M),FLB(1,M))
   12 CONTINUE
C
C
C
C     READ MODAL STIFFNESS AND DAMPING DATA
C
      CALL RTAPE(IRUNNO,KH(ICNT),KHH,NM,NM,12,12,2)
      CALL RTAPE(IRUNNO,BH(ICNT),BHH,NM,NM,12,12,2)
C
      DO 16 M=1,NM
      WRITE (6,116) M
      FLOM(M) = DSQRT(KHH(M,M))
      ZETA(M) = BHH(M,M)/(2.0*FLOM(M))
      WRITE(6,115)
      WRITE(6,123) M,FLOM(M)
      WRITE(6,115)
      WRITE(6,124) M,ZETA(M)
      WRITE(6,115)
      WRITE(6,108)  (M,(FLA(I,M),I=1,3))
      WRITE(6,115)
      WRITE(6,109)  (M,(FLB(I,M),I=1,3))
      WRITE(6,115)
      WRITE(6,110)  (M,(FLC(I,M),I=1,3))
      WRITE(6,115)
      WRITE(6,126)   (M,(FLD(1,J,M),J=1,3))
      WRITE(6,111)   (M,(FLD(I,J,M),J=1,3),I=2,3)
      WRITE(6,115)
      WRITE(6,127)   (M,(FLJ(1,J,M),J=1,3))
      WRITE(6,112)   (M,(FLJ(I,J,M),J=1,3),I=2,3)
      WRITE(6,115)
   16 CONTINUE
      WRITE (6,105)
C
C          RESULTANT COUPLED MODE DEPENDENT PARAMETERS
C
      DO 14 NN=1,NM
      DO 14 MM=1,NM
C
      DO 15 J=1,NJ
      CALL SUEOP(PHIT(1,J,NN),PHIT(1,J,MM),JMASS(J),TEM1)
      CALL DYADD(FCF(1,1,MM,NN),TEM1,FCF(1,1,MM,NN))
C
      CALL DYDOTV(JINERT(1,1,J),PHIR(1,J,MM),TEM)
      CALL VECROS(TEM,PHIR(1,J,NN),TFM)
      HF = .5
      CALL SCLV(HF,TFM,TEM)
      CALL VECROS(PHIR(1,J,MM),PHIR(1,1,NN),TFM)
      CALL SCLV(JMASS(J),TFM,TFM)
      CALL VECADD(TEM,TFM,TEM)
      CALL VECADD(FCK(1,MM,NN),TEM,FCK(1,MM,NN))
   15 CONTINUE
      WRITE (6,117) MM,NN
      WRITE(6,115)
      WRITE(6,128)   (MM,NN,(FCF(1,J,MM,NN),J=1,3))
      WRITE(6,113)   (MM,NN,(FCF(I,J,MM,NN),J=1,3),I=2,3)
      WRITE(6,115)
      WRITE(6,114)   (MM,NN,(FCK(I,MM,NN),I=1,3))
   14 CONTINUE
```

```
C
  100 FORMAT (1X,'   ** ERROR **  CANNOT FIND HINGE POINT',I5,' IN GRID P
     *OINT TABLE, RETURN OUT OF SUBROUTINE NBOD2')
  101 FORMAT (////,'   GRID POINT LABELING, LOCATION AND MASS TABLES',//)
  102 FORMAT (' IGRID(',I4,') =',I5,9X,'JLOC(',I4,') =',3E15.7,9X,
     *        'JMASS(',I4,') =',E15.7)
    5 CONTINUE
      A = 1.0/TOTMAS
      CALL SCLV(A,CMLOC,CMLOC)
C
      DO 5 N=1,NJ
      TOTMAS = TOTMAS + JMASS(N)
      CALL SCLV(JMASS(N),JLOC(1,N),TEM)
      CALL VECADD(CMLOC,TEM,CMLOC)
    8 Y(I) = X(I+1)
      NY = 3
      NR = IY(3)
      RETURN
    7 IF(IFORM.NE.307) GO TO 9
      IY(1) = NR+1
      Y(1) = X(1)
      NY = 1
      NR = X(2)*FAC
      IF(NR.EQ.0) RETURN
      IY(2) = NR
      Y(2) = X(3)
      NY = 2
      NR = X(4)*FAC
      IF(NR.EQ.0) RETURN
      NR = NR-1
      RETURN
    9 IF(IFORM.NE.308) GO TO 10
      NR = X(1)*FAC
      IF(NR.EQ.0) GO TO 14
      DO 11 I=1,3
      IY(I) = NR-1+I
   11 Y(I)  = X(I+1)
      NY = 3
      NR = IY(3)
      RETURN
   10 IF(IFORM.NE.309) GO TO 12
      NR = X(1)*FAC
      IF(NR.EQ.0) GO TO 14
      DO 13 I=1,2
      IY(I) = NR-1+I
   13 Y(I)  = X(I+1)
      NY = 2
      NR = X(4)*FAC
      IF(NR.EQ.0) RETURN
      NR = NR-1
      RETURN
  103 FORMAT ('   XMAS =',E15.7,30X,'   (TOTAL MASS OF BODY)')
  104 FORMAT (////,'  MODE SHAPE TABLE (INTERNAL NUMBERING) ',//)
  105 FORMAT (///)
  106 FORMAT ('   PHIT(',I4,',',I2,') =',3E15.7,5X,
     *      '   PHIR(',I4,',',I2,') =',3E15.7)
  107 FORMAT (///,' WARNING MASS MOMENT AT GRID POINT NON-ZERO, THAT IS
     * SMM ',I4,') =', 3E15.7,/,' THIS EFFECT IGNORED BY N-BOD2.  N-BOD2
     * ASSUMES EACH GRID POINT AT ELEMENT MASS CENTER ')
  108 FORMAT ('  FLA(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND CENTER OF
     * MASS LOCATION AFTER DEFORMATION)')
  109 FORMAT ('  FLB(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND ANGULAR M
     *OMENTUM DUE TO DEFORMATION)')
  110 FORMAT ('  FLC(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND ANGULAR M
     *OMENTUM DUE TO DEFORMATION)')
  111 FORMAT ('  FLD(',I2,') = ',3E15.7)
  112 FORMAT ('  FLJ(',I2,') = ',3E15.7)
  113 FORMAT ('  FCF(',I2,',',I2,') =',3E15.7)
  114 FORMAT ('  FCK(',I2,',',I2,') =',3E15.7,' (COEFFICIENT FOR CORIOLI
     *S TORQUE DUE TO DEFORMATION)')
  115 FORMAT ('   ')
  116 FORMAT (/,3X,'FOR MODE',I3,' THE RESULTANT UNCOUPLED MODE DEPENDEN
     *T PARAMETERS REQUIRED FOR N-BOD2 INPUT ARE')
  117 FORMAT (/,3X,'FOR MODES',I3,' AND',I3,' THE RESULTANT MODE DEPENDE
     *NT CROSS COUPLING PARAMETERS REQUIRED FOR N-BOD2 INPUT ARE')
  118 FORMAT ('1',5X,'***** RESULTANT FLEXIBLE BODY DATA REQUIRED FOR IN
     *PUT TO N-BOD2 *****',///)
  119 FORMAT ('          ',3E15.7,'  (UNDEFORMED INERTIA TENSOR OF THE')
  120 FORMAT ('  XI   =',3E15.7,'   FLEXIBLE BODY IN BODY COORDINATES')
  121 FORMAT ('          ',3E15.7,'   RELATIVE TO BODY CENTER OF MASS)')
  122 FORMAT ('  CA   =',3E15.7,'  (CENTER OF MASS VECTOR, HINGE POINT T
     *O CM)')
  123 FORMAT ('  FLOM(',I2,') =',E15.7,' RAD/SEC  (MODAL FREQUENCY)')
  124 FORMAT ('  ZETA(',I2,') =',E15.7,'           (MODAL DAMPING ZETA)')
  125 FORMAT ('  CB   = VECTOR IN CONTIGUOUS BODY TO HINGE POINT DEFINED
     * AT INTERNAL GRID POINT',I5,' (EXTERNAL GRID POINT',I5,' IN BULK D
     *ATA)')
  126 FORMAT ('  FLD(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND INERTIA T
     *ENSOR AFTER DEFORMATION)')
  127 FORMAT ('  FLJ(',I2,') = ',3E15.7,' (COEFFICIENT FOR CENTRIPITAL T
     *ORQUE DUE TO DEFORMATION)')
  128 FORMAT ('  FCF(',I2,',',I2,') =',3E15.7,' (COEFFICIENT FOR CENTRIP
     *ITAL TORQUE DUE TO DEFORMATION)')
      RETURN
      END




      SUBROUTINE INTERP(INPT.IMAT)
C
C      THIS ROUTINE SETS UP A CODE SEQUENCE WHICH WILL BE USED
C      TO READ MIXED INTEGER AND FLOATING POINT DATA FROM
C      THE DMI CARDS
C      BLANK FIELD ON LAST CARD WILL BE INTERPRETED AS AN INTEGER
C      BUT READ LATER IN AS ZERO, A CHECK FOR WHICH IS MADE
C
      INTEGER INPT(80)
      DATA IDOT,IEE/1H.,1HE/
      IF(INPT(12).EQ.IDOT.AND.INPT(21).EQ.IEE) GO TO 1
      IF(INPT(44).EQ.IDOT.AND.INPT(53).EQ.IEE) GO TO 2
      IMAT = 310
C                             310 - FORMAT  I,E,I,E
      RETURN
    2 IF(INPT(60).EQ.IDOT.AND.INPT(69).EQ.IEE) GO TO 3
      IMAT = 309
C                             309 - FORMAT  I,E,E,I
```

```
      RETURN
    3 IMAT = 308
C                              308 - FORMAT  I,E,E,E
      RETURN
    1 IF(INPT(28).EQ.IDOT.AND.INPT(37).EQ.IEE) GO TO 4
      IF(INPT(60).EQ.IDOT.AND.INPT(69).EQ.IEE) GO TO 5
      IMAT = 307
C                              307 - FORMAT  E,I,E,I  OR  E
      RETURN
    5 IMAT = 306
C                              306 - FORMAT  E,I,E,E
      RETURN
    4 IF(INPT(44).EQ.IDOT.AND.INPT(53).EQ.IEE) GO TO 6
      IMAT = 305
C                              305 - FORMAT  E,E,I,E  OR  E,E
      RETURN
    6 IF(INPT(60).EQ.IDOT.AND.INPT(69).EQ.IEE) GO TO 7
      IMAT = 304
C                              304 - FORMAT  E,E,E,I  OR  E,E,E
      RETURN
    7 IMAT = 303
C                              303 - FORMAT  E,E,E,E
      RETURN
      END
```

```
      SUBROUTINE DECODE(IFORM,X,NR,Y,IY,NY)
C
C***      ERROR IN TERMINOLOGY ELEMENTS COMING IN BY ROWS,
C            NR AND IY(I) REFER TO COLUMN NUMBER
C
C        OUTPUT OF DECODE
C             NY     = NUMBER OF REAL NUMBERS ON DMI CARD
C             IY(I)  = ROW LOCATION FOR I-TH REAL NUMBER ON DMI CARD
C             Y(I)   = I-TH REAL NUMBER
C             NR     = NEXT REAL NUMBER TO BE PUT IN ROW NR+1 UNLESS
C                      FIELD 1 OF NEXT CARD CONTAINS AN INTEGER
C
      REAL*8 X(4), Y(4)
      INTEGER  IY(4)
      FAC = 1.00001D+08
      IF(IFORM.NE.303) GO TO 1
      DO 2 I=1,4
      IY(I) = NR+I
    2 Y(I)  = X(I)
      NY = 4
      NR = IY(4)
      RETURN
    1 IF(IFORM.NE.304) GO TO 3
      DO 4 I=1,3
      IY(I) = NR+I
    4 Y(I)  = X(I)
      NY = 3
      NR = X(4)*FAC-1
      RETURN
    3 IF(IFORM.NE.305) GO TO 5
```

```
      DO 6 I=1,2
      IY(I) = NR+I
    6 Y(I)  = X(I)
      NY = 2
      NR = X(3)*FAC
      IF(NR.EQ.0) RETURN
      IY(3) = NR
      Y(3) = X(4)
      NY = 3
      RETURN
    5 IF(IFORM.NE.306) GO TO 7
      IY(1) = NR+1
      Y(1) = X(1)
      NY = 1
      NR = X(2)*FAC
      IF(NR.EQ.0) RETURN
      DO 8 I=2,3
      IY(I) = NR-2+I
   12 IF(IFORM.NE.310) GO TO 14
      NR = X(1)*FAC
      IF(NR.EQ.0) GO TO 14
      IY(1) = NR
      Y(1) = X(2)
      NR = X(3)*FAC
      IF(NR.EQ.0) RETURN
      IY(2) = NR
      Y(2) = X(4)
      NY = 2
      RETURN
   14 WRITE(6,100) IFORM
  100 FORMAT (' IFORM =',I5,' ERROR')
      STOP
      END




      SUBROUTINE INTAPE (NTAPE,TAPEID)
      REAL*8 TAPEID,BUF,EOT
      DATA IZ1,BUF,EOT/1,0.D 0,3HEOT/
      DATA NOT / 6/
C
C  INITIALIZE TAPE FOR SUBROUTINE WTAPE.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RF HRUDA.  JULY 1968.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NTAPE  = NUMBER OF TAPE. (E.G. 10).
C  TAPEID = TAPE IDENTIFICATION. (E.G. T1234). (A6 FORMAT).
C
 2001 FORMAT (//// 14H LOGICAL UNIT I2, 7H, TAPE A6,
     *               23H, HAS BEEN INITIALIZED.)
C
      REWIND NTAPE
      WRITE (NTAPE) TAPEID,IZ1,EOT,(BUF,I=1,16)
      REWIND NTAPE
      WRITE (NOT,2001) NTAPE,TAPEID
C
      RETURN
      END




      SUBROUTINE WTAPE  (A,NRA,NCA,ANAME,KR,NTAPE)
      REAL*8 A,ANAME,IRUNNO,DATE,
     *      BUF,EOT,DENSE,TAPEID,IEOTCK
      DIMENSION  A(KR,1)
      DATA IRUNNO,DATE/6HPREPRS,6HDATE  /
      DATA BUF,EOT,DENSE/0.D 0,3HEOT,5HDENSE/
      DATA NOT / 6/
C
C  WRITE MATRIX A ON TAPE.
C  INITIALIZE TAPE WITH SUBROUTINE INTAPE.
C  REWIND TAPE BEFORE FIRST USE OF THIS SUBROUTINE.
C  NOTE...THIS ROUTINE IS DESIGNED SPECIFICALLY FOR WRITING ON A DISK
C
C         (EG CDC-6400 DISK). USING THIS ROUTINE TO WRITE ON A PHYSICAL
C         TAPE DIRECTLY (IE WITHOUT USING THE DISK AS AN INTERMEDIARY)
C         WILL PROBABLY GIVE POOR RESULTS (DUE TO THE TOLERANCE
C         CHARACTERISTICS OF A TAPE DRIVE) AND SHOULD BE AVOIDED IF AT
C         ALL POSSIBLE.
C      ...THE CDC-6400 DISK IS AUTOMATICALLY ENDFILED AFTER EACH WRITE.
C  CODED BY   W A BENFIELD.  MARCH 1966.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C  REVISED BY  RF HRUDA.  NOVEMBER 1970.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  A      = MATRIX TO BE WRITTEN ON TAPE. SIZE(NRA,NCA).
C  NRA    = NUMBER OF ROWS OF MATRIX A.
C  NCA    = NUMBER OF COLS OF MATRIX A.
C  ANAME  = MATRIX IDENTIFICATION. (A6 FORMAT).
C  KR     = ROW DIMENSION OF A IN CALLING PROGRAM.
C  NTAPE  = NUMBER OF TAPE. (E.G. 10).
C
C     INTERNAL VARIABLES THAT ARE PUT ON TAPE (TRANSFERRED THRU COMMON).
C  IRUNNO IS RUN NUMBER OF PROBLEM. (A6 FORMAT).
C  DATE IS DATE. (A6 FORMAT).  FOR EXAMPLE  15FE65.
C
      IF (NRA .LT. 1 .OR. NCA .LT. 1) GO TO 999
C
C  SEARCH TAPE FOR END OF WRITTEN DATA.
   10 READ (NTAPE) TAPEID,LN,IEOTCK
      IF (IEOTCK .EQ.   EOT) GO TO 20
      READ (NTAPE)
      GO TO 10
C
C  END OF WRITTEN DATA HAS BEEN FOUND.
   20 BACKSPACE NTAPE
      WRITE (NTAPE) TAPEID,LN,BUF,IRUNNO,ANAME,NRA,NCA,DATE,DENSE,
     *              (BUF,I=1,10)
      WRITE (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
      LN = LN + 1
      WRITE (NTAPE) TAPEID,LN,EOT,(BUF,I=1,16)
      BACKSPACE NTAPE
      RETURN
C
  999 WRITE (NOT,1000)
 1000 FORMAT (1H1,43HERROR IN SUBROUTINE WTAPE, PROGRAM STOPPED.)
      STOP
      END
```

```
      SUBROUTINE LTAPE (NTAPE)
      REAL*8 TAPEID,IRUNNO,ANAME,IEOTCK,DATE,ITYPE,ICHK,EOT,
     *    DENSE,SPARSE,SPART
          DATA NOT / 6/
          DATA EOT,DENSE,SPARSE,SPART/ 3HEOT,5HDENSE,6HSPARSE,5HSPART/
C
C  LIST HEADINGS OF MATRICES ON TAPE.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RF HRUDA.  JULY 1968.   REVISED NOVEMBER 1970.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NTAPE  = NUMBER OF TAPE. (E.G. 10).
C
 2001 FORMAT (//36X35HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6)
 2002 FORMAT (//30X35HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6,
     *         12H (CONTINUED))
 2003 FORMAT (27X69(1H-)/27X3HNO.3X7HRUN NO.4X4HNAME5X5HNROWS4X5HNCOLS4X
     *         4HDATE6X3HNNZ3X9HPARTITION/
     *           27X3H---3X6H------4X6H------4X5H-----
     *           4X5H-----3X6H------5X3H---3X9H--------- / )
 2004 FORMAT (25XI5,3XA6,4XA6,3XI5,4XI5,4XA6,3XI5,3XI4,1H/I4)
 2005 FORMAT (/27X12HEND OF LIST.)
C
      REWIND NTAPE
      READ  (NTAPE) TAPEID
      REWIND NTAPE
      L=0
C
   12 CONTINUE
      IF(L .EQ. 0)  WRITE (NOT,2001) NTAPE,TAPEID
      IF(L .NE. 0)  WRITE (NOT,2002) NTAPE,TAPEID
      WRITE (NOT,2003)
      NLINE=1
   13 L=L+1
      READ (NTAPE) TAPEID,LN,IEOTCK,IRUNNO,ANAME,NR,NC,DATE,ITYPE,NNZ,
     *               NP,NPT
      IF (L .EQ. 1)  ICHK = IRUNNO
      IF (ICHK .EQ. IRUNNO) GO TO 15
      NLINE=NLINE+1
      WRITE (NOT,2004)
      ICHK = IRUNNO
   15 IF (IEOTCK .EQ. EOT) GO TO 30
      READ  (NTAPE)
      IF (ITYPE .EQ.   DENSE ) WRITE (NOT,2004)
     *                           LN,IRUNNO,ANAME,NR,NC,DATE
```

```
      IF (ITYPE .EQ.   DENSE ) GO TO 20
      IF (ITYPE .EQ.   SPARSE) WRITE (NOT,2004)
     *                         LN,IRUNNO,ANAME,NR,NC,DATE,NNZ
      IF (ITYPE .EQ.   SPARSE) GO TO 20
      IF (ITYPE .EQ.   SPART ) WRITE (NOT,2004)
     *                         LN,IRUNNO,ANAME,NR,NC,DATE,NNZ,NP,NPT
      IF (ITYPE .EQ.   SPART ) GO TO 20
      WRITE (NOT,2004) LN,IRUNNO,ANAME,NR,NC,ITYPE
   20 NLINE=NLINE+1
      IF(NLINE.GT.43) GO TO 12
      GO TO 13
C
   30 WRITE (NOT,2004) LN,IEOTCK
      WRITE (NOT,2005)
      REWIND NTAPE
      RETURN
      END
```

```
      SUBROUTINE RTAPE  (IARUNO,IANAME, A,NRA,NCA, KR,KC,NTAPE)
      REAL*8 A,IARUNO,IANAME,TAPEID,IEOTCK,ITRUNO,ITNAME,
     *       DATE,ITYPE,DENSE,EOT
      DIMENSION A(KR,1)
      DATA NOT / 6 /
      DATA DENSE,EOT / 5HDENSE,3HEOT /
C
C  READ MATRIX A FROM TAPE BY IDENTIFICATION OF IARUNO,IANAME.
C  CALLS FORMA SUBROUTINES LTAPE,PAGEHD,ZZBOMB.
C  CODED BY WA BENFIELD.  JUNE 1966.
C  LAST REVISION BY R F HRUDA.  SEPTEMBER 1971.
C
C     SUBROUTINE ARGUMENTS
C  IARUNO = INPUT  RUN NUMBER OF MATRIX A. (A6 FORMAT).
C  IANAME = INPUT  MATRIX IDENTIFICATION.  (A6 FORMAT).
C  A      = OUTPUT MATRIX READ FROM TAPE. SIZE(NRA,NCA).
C  NRA    = OUTPUT NUMBER OF ROWS OF MATRIX A. WILL BE READ FROM TAPE.
C  NCA    = OUTPUT NUMBER OF COLS OF MATRIX A. WILL BE READ FROM TAPE.
C  KR     = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C  KC     = INPUT  COL DIMENSION OF A IN CALLING PROGRAM.
C  NTAPE  = INPUT  NUMBER OF TAPE. (E.G. 10).
C
 3001 FORMAT (29H1RTAPE CANNOT FIND   RUNNO = A6 /
     *        21X 8HANAME = A6 / 29X 6H------ )
C
      NTIME = 0
C  SEARCH TAPE FOR CORRECT HEADING.
    5 READ (NTAPE) TAPEID,LN,IEOTCK,ITRUNO,ITNAME,NRA,NCA,DATE,ITYPE,NNZ
      IF (ITRUNO .EQ. IARUNO .AND. ITNAME .EQ. IANAME) GO TO 10
      IF (IEOTCK .EQ.   EOT) GO TO 20
      READ  (NTAPE)
      GO TO 5
C
C  MATRIX HAS BEEN FOUND.
   10
```

```
                                                           NERROR=1
      IF (ITYPE .NE.   DENSE  .AND. NNZ .NE. 0) GO TO 999
                                                           NERROR=2
      IF (NRA.GT.KR .OR. NCA.GT.KC)  GO TO 999
      READ  (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
      RETURN
C
C  MATRIX CANNOT BE FOUND. SEARCH TAPE ONCE MORE.
   20 NTIME = NTIME+1
                                                           NERROR=3
      IF (NTIME .EQ. 2) GO TO 998
      REWIND NTAPE
      GO TO 5
C
  998 WRITE (NOT,3001) IARUNO,IANAME
  999 CALL LTAPE  (NTAPE)
      WRITE (NOT,3002) NERROR
 3002 FORMAT (1H1,43HERROR IN SUBROUTINE RTAPE, PROGRAM STOPPED,
     *        10H NERROR = ,I3)
      STOP
      END




      SUBROUTINE WRITE  (A,NR,NC,ANAME,KR)
      REAL*8 A,ANAME
      DIMENSION A(KR,1)
      DATA NOT / 6/
C
C  WRITE MATRIX OF REAL NUMBERS ON PAPER.
C  REQUIRES 123 COLUMN (MINIMUM) PRINTER.
C  UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RL WOHLEN.  DECEMBER 1968.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  A     = MATRIX TO BE PRINTED. SIZE(NR,NC).
C  NR    = NUMBER OF ROWS IN MATRIX A.
C  NC    = NUMBER OF COLS IN MATRIX A.
C  ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C  KR    = ROW DIMENSION OF A IN CALLING PROGRAM.
C
 2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //
     *        10X,10(7X,1H( I2,1H))/)
 2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H )
     *         3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/)
 2030 FORMAT (1X,2I5,2X,1P10D11.3)
 2040 FORMAT (14H0END OF WRITE.)
C
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.
      WRITE (NOT,2010) ANAME,NR,NC,(L,L=1,10)
      NLINE = 0
C
      DO 60 I=1,NR
      NZERO = 0
      JS = 1
   10 JF = JS+9
```

```
      IF (JE .GT. NC) JE=NC
C   SEE IF ELEMENTS ARE ZERO.
      DO 20 J=JS,JE
      IF (A(I,J) .NE. 0.D 0) GO TO 30
   20 CONTINUE
      GO TO 40
   30 NLINE = NLINE+1
      IF (NLINE .LE. 44) GO TO 35
      WRITE (NOT,2020) ANAME,NR,NC,(L,L=1,10)
      NLINE = 1
   35 WRITE (NOT,2030) I,JS,(A(I,J), J=JS,JE)
      NZERO = 1
   40 IF (JE .EQ. NC) GO TO 50
      JS = JS+10
      GO TO 10
C   SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS
C   AND SOMETHING HAS BEEN WRITTEN.
   50 IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR)  GO TO 60
      NLINE = NLINE+1
      WRITE (NOT,2030)
   60 CONTINUE
C
      WRITE (NOT,2040)
      RETURN
      END




      SUBROUTINE VECADD(V1,V2,S)
C     ADDS VECTOR V1 TO V2 RESULT IN S
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3), S(3)
      S(1) = V1(1) + V2(1)
      S(2) = V1(2) + V2(2)
      S(3) = V1(3) + V2(3)
      RETURN
      END




      SUBROUTINE VECSUB(V1,V2,D)
C     SUBTRACTS VECTORS V1-V2=D
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),D(3)
      D(1) = V1(1) - V2(1)
      D(2) = V1(2) - V2(2)
      D(3) = V1(3) - V2(3)
      RETURN
      END
      SUBROUTINE SCLV(SC,V,P)
C     SCALAR * VECTOR
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V(3),P(3)
      P(1) = SC*V(1)
      P(2) = SC*V(2)
      P(3) = SC*V(3)
      RETURN
      END
```

```
      SUBROUTINE VECDOT(V1,V2,D)
C     VECTOR DOT PRODUCT
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3)
      D = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
      RETURN
      END
```

```
      SUBROUTINE VECROS (V1,V2,C)
C     VECTOR CROSS PRODUCT  C = V1 X V2
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),C(3)
      C(1) = V1(2)*V2(3)- V1(3)*V2(2)
      C(2) = V1(3)*V2(1)- V1(1)*V2(3)
      C(3) = V1(1)*V2(2)- V1(2)*V2(1)
      RETURN
      END
```

```
      SUBROUTINE TRIPVP(V1,V2,V)
C
C      COMPUTES STANDARD VECTOR TRIPLE PRODUCT
C
C                    V = V1X(V1XV2)
C                      = V1*(V1.V2) - V2*(V1.V1)
C
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),V(3)
      A = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
      B = V1(1)*V1(1) + V1(2)*V1(2) + V1(3)*V1(3)
      V(1) = V1(1)*A - V2(1)*B
      V(2) = V1(2)*A - V2(2)*B
      V(3) = V1(3)*A - V2(3)*B
      RETURN
      END
      SUBROUTINE DYADD(D1,D2,D)
C     ADDS TWO DYADS
C             D = D1 + D2
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D1(3,3), D2(3,3), D(3,3)
      DO 1  I=1,3
      DO 1  J=1,3
    1 D(I,J) = D1(I,J) + D2(I,J)
      RETURN
      END
```

```
      SUBROUTINE SCLD(A,D,T)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D(3,3),T(3,3)
C     MULTIPLY SCALAR BY A TENSOR
      T(1,1) = A*D(1,1)
      T(2,1) = A*D(2,1)
      T(3,1) = A*D(3,1)
      T(1,2) = A*D(1,2)
      T(2,2) = A*D(2,2)
      T(3,2) = A*D(3,2)
      T(1,3) = A*D(1,3)
      T(2,3) = A*D(2,3)
      T(3,3) = A*D(3,3)
      RETURN
      END
```

```
      SUBROUTINE DYDOTV(A,V,D)
C     SCALAR DOT PRODUCT OF DYAD AND VECTOR
C                  D = A.V
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D(3),A(3,3),V(3)
      DO 1  I=1,3
      D(I) = 0
      DO 1  J=1,3
    1 D(I) = D(I) + A(I,J)*V(J)
      RETURN
      END
```

```
      SUBROUTINE VXDYDV(V1,DY,V)
C     COMPUTES   VECTOR X (DYAD . VECTOR)
C             V = V1 X (DY . V1)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),V(3),DY(3,3)
      DO 1 K=1,3
      V2(K) = 0.D0
      DO 1  J=1,3
    1 V2(K) = V2(K) + DY(K,J)*V1(J)
      V(1)= V1(2)*V2(3) - V1(3)*V2(2)
      V(2) = V1(3)*V2(1) - V1(1)*V2(3)
      V(3) = V1(1)*V2(2) - V1(2)*V2(1)
      RETURN
      END




      SUBROUTINE DYTOV  (D,X1,X)
C     USE TO TAKE SCALAR DOT PRODUCT OF TRANSPOSE OF
C      TENSOR D WITH VECTOR X1
C     NEEDED SINCE TENSORS IN SYMMERIC MATRIX OF INERTIA TENSORS ARE IN
C        NON SYMMETRIC
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D(3,3),X1(3),X(3)
      DO 1  I=1,3
      X(I) = 0
      DO 1  J=1,3
      X(I) = X(I) + D(J,I)*X1(J)
    1 CONTINUE
      RETURN
      END




      SUBROUTINE VODYOV(V1,DY,V2,X)
C      COMPUTES THE SCALAR TRIPLE PRODUCT
C                    VECTOR . (DYAD . VECTOR)
C                          V1.(DY.V2)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),DY(3,3),V2(3),TEM(3)
      DO 1  K=1,3
      TEM(K) = 0.D0
      DO 1  J=1,3
    1 TEM(K) = TEM(K) + DY(K,J)*V2(J)
      X = 0
      DO 2  J=1,3
    2 X = X + V1(J)*TEM(J)
      RETURN
      END




      SUBROUTINE DYOP(V,D)
C     TRANSFORMS VECTOR V1 INTO SKEW DYAD
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION  V(3),D(3,3)
      D(1,1) = 0
      D(1,2) = V(3)
      D(1,3) = -V(2)
      D(2,1) = -V(3)
      D(2,2) = 0
      D(2,3) = V(1)
      D(3,1) = V(2)
      D(3,2) = -V(1)
      D(3,3) = 0
      RETURN
      END



      SUBROUTINE SUEDP(V1,V2,XM,D)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),D(3,3)
C     USED TO COMPUTE THE PSUEDO INERTIA TENSOR
C      OF BODY LAMBA WITH RESPECT TO THE ORIGIN OF NEST K-1 AND
C      THE HINGE POINT I-1 WHICH IS ON THE TOPOLOGICAL PATH FROM
C      BODY 1 TO BODY LAMBA
C          BLOCK G SUPPER GAMBA,SUB K-1,I-1 EQUATION 2-55 OF X-732-71-70
C                        D = XM*((V1.V2)*1 - V2 V1)
C
C         XM - SCALAR
C         V1 - VECTOR
C         V2 - VECTOR
C          1 - UNIT DYAD
C          * - SCALAR MULTIPLICATION
C          . - VECTOR SCALAR MULTIPLICATION
C     BLANK - TENSOR MULTIPLICATION
C      NOTE THAT IN GENERAL THE PSUEDO INERTIA TENSOR IS NON SYMMETRIC
C
      D(1,1) = XM*(V1(2)*V2(2) + V1(3)*V2(3))
      D(1,2) =-XM*V2(1)*V1(2)
      D(1,3) =-XM*V2(1)*V1(3)
      D(2,1) =-XM*V2(2)*V1(1)
      D(2,2) = XM*(V2(1)*V1(1) + V2(3)*V1(3))
      D(2,3) =-XM*V2(2)*V1(3)
      D(3,1) =-XM*V2(3)*V1(1)
      D(3,2) =-XM*V2(3)*V1(2)
      D(3,3) = XM*(V2(1)*V1(1) + V2(2)*V1(2))
      RETURN
      END



      SUBROUTINE VECXDY(P,T,D)
      REAL*8  P(3),  T(3,3),  D(3,3)
C
C        COMPUTES VECTOR CROSS DYAD
C                    D = P X T
      D(1,1) = P(2)*T(3,1) - P(3)*T(2,1)
      D(1,2) = P(2)*T(3,2) - P(3)*T(2,2)
      D(1,3) = P(2)*T(3,3) - P(3)*T(2,3)
      D(2,1) = P(3)*T(1,1) - P(1)*T(3,1)
      D(2,2) = P(3)*T(1,2) - P(1)*T(3,2)
      D(2,3) = P(3)*T(1,3) - P(1)*T(3,3)
      D(3,1) = P(1)*T(2,1) - P(2)*T(1,1)
      D(3,2) = P(1)*T(2,2) - P(2)*T(1,2)
      D(3,3) = P(1)*T(2,3) - P(2)*T(1,3)
      RETURN
      END
```

Goddard Space Flight Center
National Aeronautics and Space Administration
Greenbelt, Maryland October 1977

APPENDIX A

INPUT/OUTPUT SUBROUTINE EXPLANATIONS

BLANK PAGE

BLANK PAGE

APPENDIX A

INPUT/OUTPUT SUBROUTINE EXPLANATIONS

This appendix presents excerpts from *Synthesis of Dynamic Systems Using FORMA – Fortran Matrix Analysis*, MCR-71-75, Martin Marietta Corporation, Denver, Colorado, May 1971, that explain the subroutines from the FORMA library used in the digital computer program.

COMENT

Subroutine COMENT reads input comment cards and reproduces each card in the printed output of the computer run. Each comment card may have any keypunch symbol in card columns 1 through 78. A use of COMENT is to print an explanation of coordinates used in a computer run. Thus, this information is always retained with a run to correlate matrix location numbers with physical coordinates.

INTAPE

Subroutine INTAPE initializes a tape* for the FORMA tape system by writing EOT (end of tape) at the beginning of the tape (disk). All FORMA tape subroutines recognize the EOT as the end of written data. Each "new" tape (disk) must be initialized with this subroutine INTAPE to make the tape (disk) compatible with the other FORMA tape subroutines (LTAPE, RTAPE, WTAPE, and UPDATE).

A "new" tape (disk) is defined as a tape (disk) for which it is desired to begin writing matrix data at the front of the tape (disk). Thus, a "new" tape (disk) could be one that contains obsolete FORMA matrix data, as well as one on which the FORMA system has never written.

As an example, pertinent statements from a program containing INTAPE could be:

```
       DATA  NIT,NOT/5,6/
1001   FORMAT   (12A6)
       NRTAPE =  10
       READ  (NIT, 1001) IFINIT,   TAPEID
       IF (IFINIT   .EQ. 6HINITIL)  CALL INTAPE(NRTAPE,TAPEID)
                 .
                 .
                 .
```

---

*A disk is preferred. See writeup of subroutine WTAPE at the end of this appendix.

The input data (beginning in card column 1) to this example program would be either of the following:

- INITILTXXXX, if the tape is to be initialized. (TXXXX represents the particular tape number used; (e.g., T1234).
- NOINIT, if the tape is *not* to be initialized. The tape identification is not needed.

LTAPE

Subroutine LTAPE lists the matrix headings (see subroutine WTAPE writeup) written on a FORMA tape (or disk). The following matrix headings were written by subroutine WTAPE and consist of:

NO. = Matrix number on tape

RUN NO. = Run number of problem when matrix was written on tape

NAME = Matrix name

NROWS = Number of rows of matrix

NCOLS = Number of columns of matrix

DATE = Date when matrix was written on tape

NNZ = Number of nonzeros (used only in sparse FORMA in which only nonzeros are used)

PARTITION = Partition number of sparse matrix

READ

Subroutine READ reads a matrix of real numbers (a FORTRAN term for numbers with a decimal point) from either cards or tape into the computer. The matrix is then printed so that these input data are recorded with the answers of a run. A print suppression option is available for a matrix read from tape. On option, the matrix read from either cards or tape may be written on a tape (by subroutine WTAPE).

The first data card read by subroutine READ contains the information for indicating whether cards or tape will be used. The information entered on this card (and subsequent cards for card input) is as follows:

Card Data Input Form

In the following, an asterisk denotes required entries. Other entries are optional.

	Card Columns	Format Type[1]		Entry
First Card	1-6	A		*Matrix name—will appear in printout
	7-10	I		*Matrix row size
	11-15	I		*Matrix column size
	16-69	A		Any remarks to further identify the input matrix
Write-Tape Options	72			$.—Only if the Write-Tape is to be initialized by subroutine INTAPE. The Write-Tape identification will be from card columns 73 through 78.
	72		or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A		Write-Tape identification (e.g., T1234)—use with $ in card column 72.
	73-78		or	REWIND—The Write-Tape will be rewound before being used.
	73-76		or	LIST—The Write-Tape will be listed by subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78		or	Anything else will be ignored.
	79-80	I		Write-Tape number (e.g., 21)
			or	Blank if the matrix is not to be written on tape.
Middle Cards	1-5	I		*Row number of matrix elements on card
	6-10	I		*Column number of matrix element in first data field.

	Card Columns	Format Type[1]	Entry
Middle Cards (continued)	11-27	E	*First data field with matrix elements[2]
	28-44	E	*Second data field with matrix elements[2]
	45-61	E	*Third data field with matrix elements[2]
	62-78	E	*Fourth data field with matrix elements[2]
Last Card	1-10	I	*Ten zeroes

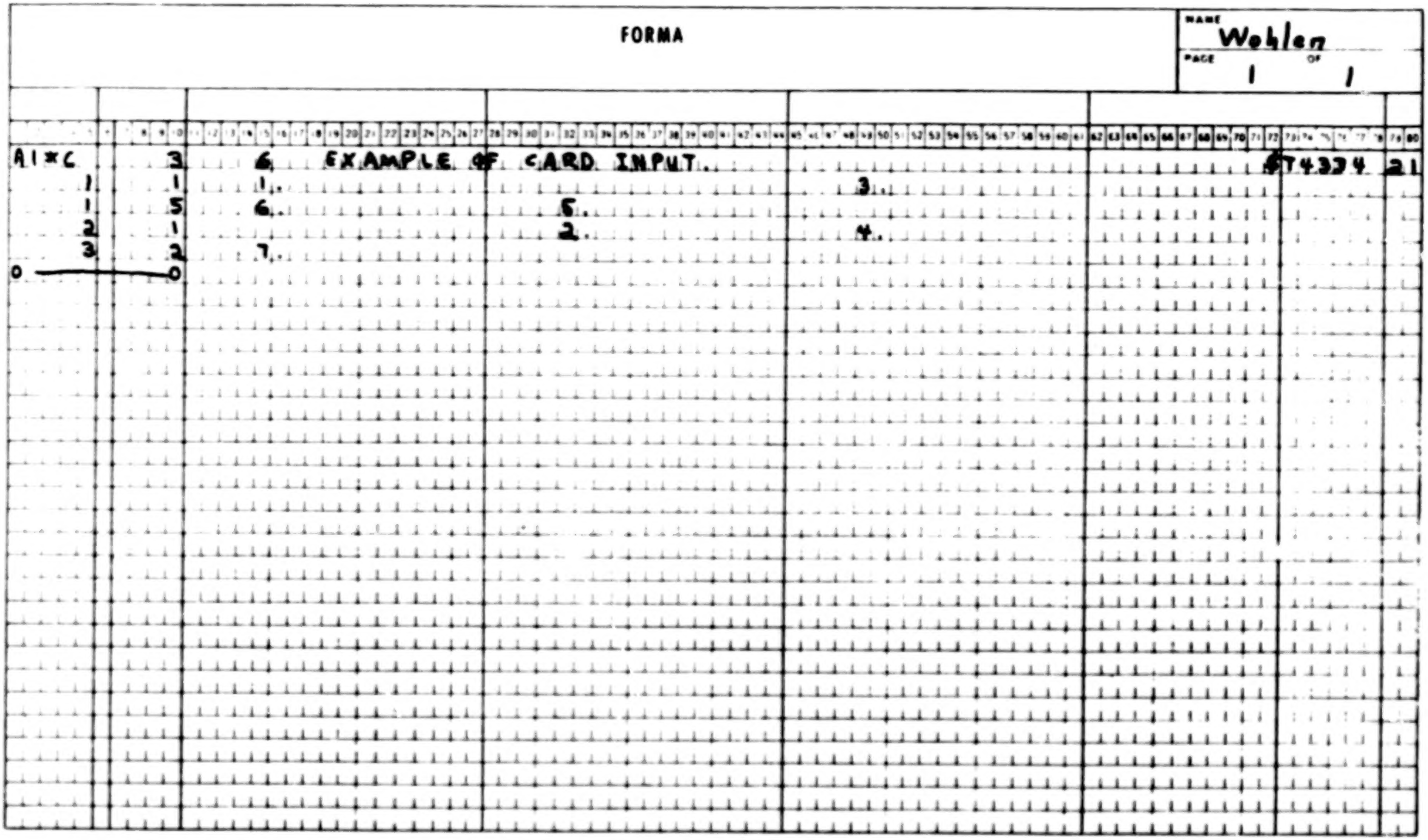
Note 1: Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right-justified in the field. Format Type E allows only read numbers (a FORTRAN term for numbers with a decimal point) anywhere in the field).

Note 2: Only nonzero elements need be entered.

As an example of card input to subroutine READ, consider the following matrix:

$$[A1*C]_{3\times 6} = \begin{bmatrix} 1. & 0. & 3. & 0. & 6. & 5. \\ 0. & 2. & 4. & 0. & 0. & 0 \\ 0. & 7. & 0. & 0. & 0. & 0. \end{bmatrix}$$

This matrix is also to be written on tape 21 that is to be initialized and identified as T4334. Figure A-1 demonstrates how this information could be written on a coding form to facilitate keypunching to cards.



FORMA

NAME Wohlen

PAGE 1 OF 1

```
A1*C     3    6   EXAMPLE OF CARD INPUT.                               6T4334 21
    1    1    1.                                3.
    1    5    6.               5.
    2    1                     2.               4.
    3    2    7.
0--------0
```

Figure A-1. Example of card input for subroutine READ.

Tape Data Input Form

In the following, an asterisk denotes required entries. Other entries are optional. Only one card is used for each matrix read.

	Card Columns	Format Type[1]		Entry
One Card	1-6	A		*Name of matrix to be read from Read-Tape.
	10			Zero – Read-Tape will move forward from its present position and search to the end of the tape. If the matrix is not found on the first end-of-tape encounter, the tape will automatically rewind and make one more pass. If it is not found on the second end-of-tape encounter, an error message will be printed and the program will stop.
	7-10	I	or	Minus the location number of matrix on the Read-Tape. Tape will be positioned at the beginning of the location specified and will then continue as described above for a zero in column 10.
	11-15	I		*Read-Tape number (e.g., 11) – If positive, the matrix read will be printed in the output. If negative, the matrix read will not be printed in the output.
	16-21	A		*Run number of matrix to be read from the Read-Tape.
	22-27			REWIND – Read-Tape will be rewound before being used.
	22-25		or	LIST – Read-Tape will be listed by subroutine LTAPE.

	Card Columns	Format Type[1]		Entry
	22-27		or	Anything else will be considered as part of the remarks described below.
	28-69	A		Any remarks to further identify the input matrix.
Write-Tape Options	72			$.–Only if the Write-Tape is to be initialized by subroutine INTAPE. The Write-Tape identification will be from card columns 73 through 78.
	72		or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A		Write-Tape identification (e.g., T1234)–Use with $ in card column 72.
	73-78		or	REWIND–Write-Tape will be rewound before being used.
	73-76		or	LIST–Write-Tape will be listed by subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78		or	Anything else will be ignored.
	79-80	I		Write-Tape number (e.g., 21)
			or	Blank if the matrix is not to be written on tape.

Note 1: Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right-justified in the field.

As examples of tape input to subroutine READ, consider:

- Example 1–A matrix named AB2 with run number of RUN-46 is to be read from tape number 11 into the computer and printed. This matrix is also to be written on tape number 22 that is to be initialized and identified as T4321.

- Example 2—A matrix named XYZ4 with run number of TKD is on tape number 13 twice—the first time at location 29 and the second time at location 54. It is desired to read the second matrix.

Figure A-2 demonstrates how these two examples would be written on a coding form to facilitate keypunching to cards.

READIM

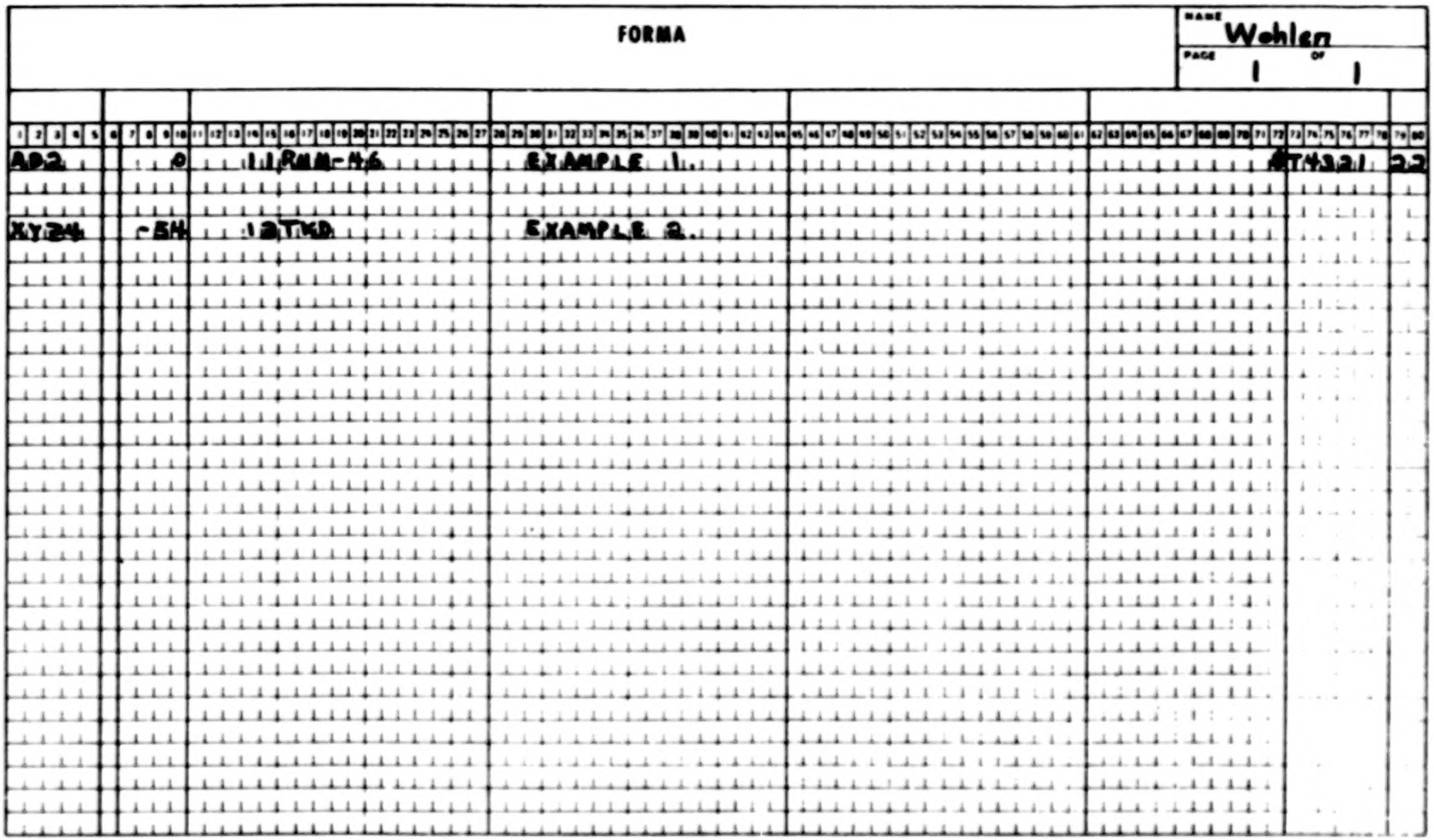
Subroutine READIM reads a matrix of integer numbers from either cards or tape into the computer. The matrix is then printed so that these input data are recorded with the answers of a run. A print suppression option is available for a matrix read from tape. On option, the matrix read from either cards or tape may be written on a tape (by subroutine WTAPE).

The first data card read by subroutine READIM contains the information to indicate whether cards or tape will be used. The information entered on this card (and subsequent cards for card input) is given below.

Card Data Input Form

Required entries are denoted by an asterisk below. Any other entry is optional.

	Card Columns	Format Type[1]		Entry
First Card	1-6	A		*Matrix name—Will appear in printout
	7-10	I		*Matrix row size
	11-15	I		*Matrix column size
	16-69	A		Any remarks to further identify the input matrix
Write-Tape Options	72			$—Only if the Write-Tape is to be initialized by subroutine INTAPE. The Write-Tape identification will be from card columns 73 through 78.
	72		or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A		Write-Tape identification (e.g., T1234)—Use with $ card column 72.



FORMA

NAME Wehlen

PAGE 1 OF 1

```
AB2      0   11RMM-46        EXAMPLE 1.                                AT4321 22
XY24   -5H    12TKD          EXAMPLE 2.
```

Figure A-2. Examples of tape input for subroutine READ.

	Card Columns	Format Type[1]		Entry
	73-78		or	REWIND–Write-Tape will be rewound before being used.
	73-76		or	LIST–Write-Tape will be listed by subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78		or	Anything else will be ignored.
	79-80	I		Write-Tape number (e.g., 21)
			or	Blank if the matrix is not to be written on tape.
Middle Cards	1-5	I		*Row number of matrix elements on card
	6-10	I		*Column number of matrix element in first data field
	11-15	I		*First data field with matrix elements[2]
	16-20	I		*Second data field with matrix elements[2]
	etc.			
	76-80	I		*Fourteenth data field with matrix elements[2]
Last Card	1-10	I		*Ten zeroes

Note 1: Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right-justified in the field.

Note 2: Only nonzero elements need be entered.

As an example of card input to subroutine READIM, consider the following matrix:

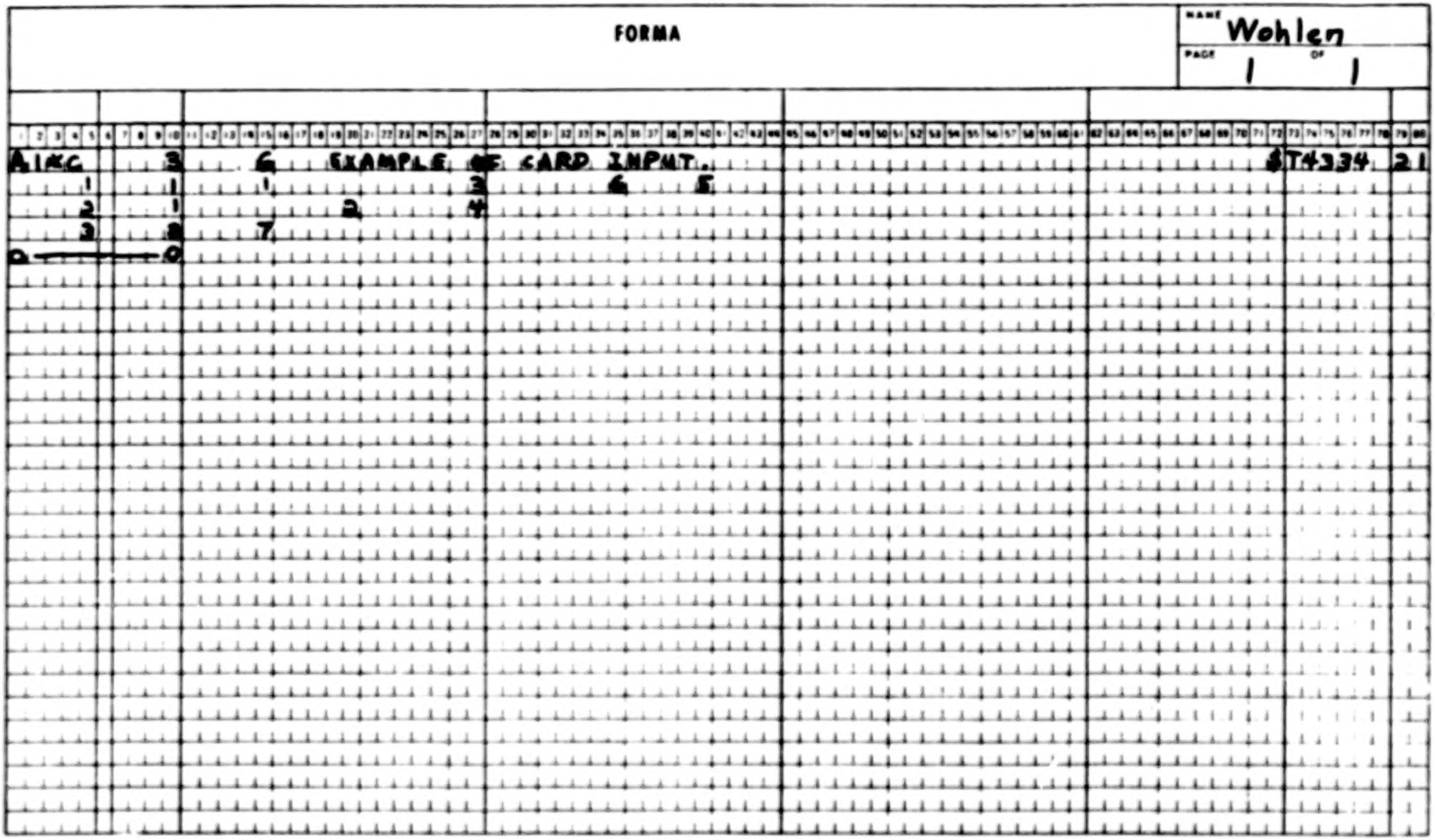
$$[A1{*}C]_{3\times 6} = \begin{bmatrix} 1 & 0 & 3 & 0 & 6 & 5 \\ 0 & 2 & 4 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This matrix is also to be written on tape 21 that is to be initialized and identified as T4334. Figure A-3 demonstrates how this information could be written on a coding form to facilitate keypunching to cards.

Tape Data Input Form

In the following, an asterisk denotes required entries. Other entries are optional. Only one card is used for each matrix read.

	Card Columns	Format Type[1]		Entry
One Card	1-6	A		*Name of matrix to be read from the Read-Tape
	10			Zero—Read-Tape will move forward from its present position and search to the end of the tape. If the matrix is not found on the first end-of-tape encounter, the tape will automatically rewind and make one more pass. If it is not found on the second end-of-tape encounter, an error message will be printed and the program will stop.
	7-10	I	or	Minus the location number of matrix on the Read-Tape. Tape will be positioned at the beginning of the location specified and will then continue as described above for a zero in column 10.



FORMA

NAME Wohlen

PAGE 1 OF 1

```
AIKG     3    6   EXAMPLE OF CARD INPUT.                               $T4334 21
    1    1    1           3       6    5
    2    1         2      4
    3    8    7
0--------0
```

Figure A-3. Example of card input for subroutine READIM.

	Card Columns	Format Type[1]		Entry
One Card	11-15	I		*Read-Tape number (e.g., 11)–If positive, the matrix read will be printed in the output. If negative, the matrix read will not be printed in the output.
	16-21	A		*Run number of matrix to be read from the Read-Tape.
	22-27			REWIND–Read-Tape will be rewound before being used.
	22-25		or	LIST–Read-Tape will be listed by subroutine LTAPE.
	22-27		or	Anything else will be considered as part of the remarks described below.
	28-69	A		Any remarks to further identify the input matrix.
Write-Tape Options	72			$–Only if the Write-Tape is to be initialized by subroutine INTAPE. The Write-Tape identification will be from card columns 73 through 78.
	72		or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A		Write-Tape identification (e.g., T1234)–Use with $ in card column 72.
	73-78		or	REWIND–Write-Tape will be rewound before being used.
	73-76		or	LIST–Write-Tape will be listed by subroutine LTAPE after the matrix has been written on the Write-Tape.

Card Columns	Format Type[1]		Entry
73-78		or	Anything else will be ignored.
79-80	I		Write-Tape number (e.g., 21)
		or	Blank if the matrix is not to be written on tape.

Note 1: Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right-justified in the field.

As examples of tape input to subroutine READIM, consider:

- Example 1–A matrix named AB2 with run number of RUN-46 is to be read from tape number 11 into the computer and printed. This matrix is also to be written on tape number 22 that is to be initialized and identified as T4321.
- Example 2–A matrix named XYZ4 with run number of TKD is on tape number 13 twice. The first time is at location 29 and the second time is at location 54. It is desired to read the second matrix.

Figure A-4 demonstrates how these two examples would be written on a coding form to facilitate keypunching to cards.

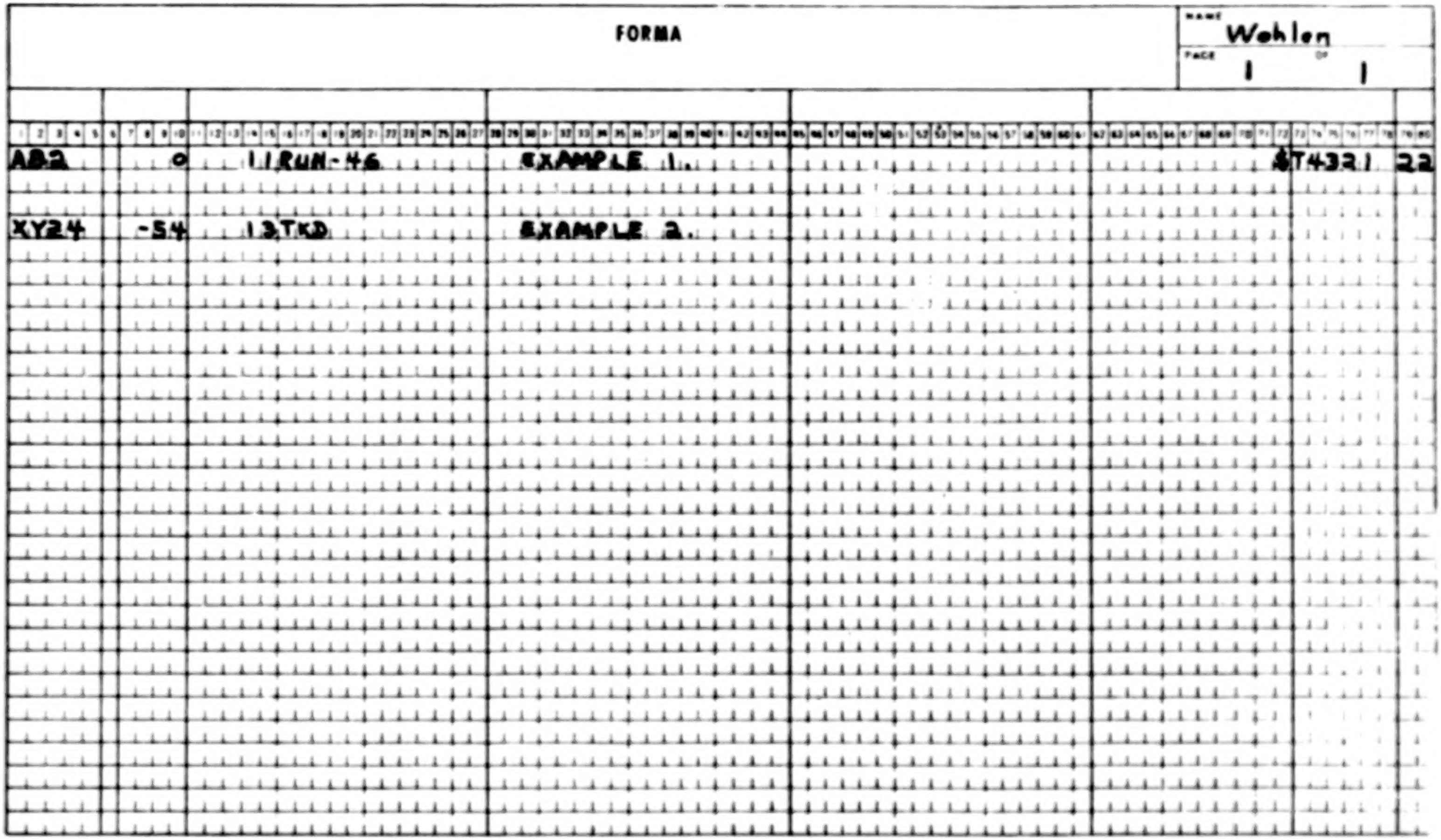
RTAPE

Subroutine RTAPE reads a selected matrix from tape (disk) into the computer core. The matrix to be selected is identified by the desired run number and matrix name. This procedure is accomplished by searching the matrix headings (see subroutine WTAPE writeup) until a match with the desired run number and matrix name is obtained and then by reading the matrix elements from tape (disk) into the computer core. The search begins at the current position (does not rewind) of the tape (disk) and proceeds to the EOT (end of tape defined in subroutine WTAPE writeup). If the desired matrix was not found on reaching the EOT, a rewind is performed, and one more search to the EOT is made. If the desired matrix is again not found: (1) an error message is printed, (2) a listing of the matrix headings is printed (see subroutine LTAPE writeup), and (3) transfer is made to subroutine ZZBOMB where the program is terminated.

START

Subroutine START performs the following operations:

- Reads input card 1 for the run number (any keypunch symbol in card columns 1 through 6) and the user's name (any keypunch symbol in card columns 11 through 28).



FORMA

NAME Wohlen

PAGE 1 OF 1

```
AB2      0  11RUN-46        EXAMPLE 1.                                 $T4321 22


XY24   -54  13TKD           EXAMPLE 2.
```

Figure A-4. Examples of tape input for subroutine READIM.

CONTENTS

Page

PREFACE . . . . iii 1/A5

I. PROGRAM SYSTEM OVERVIEW 1 1/A7

A. Introduction 2 1/A8
B. Simulation Overview and Nomenclature 3 1/A9
C. General Use Information 8 1/A14

II. PROGRAM SEGMENTATION 9 1/B1

III. DELINEATION OF USER-SUPPLIED MODULES 10 1/B2

A. Logic Flow 11 1/B3
B. User-Supplied Module Specifics 12 1/B4
C. Selected Common-Block Information 17 1/B9

IV. PROGRAM INPUTS 19 1/B11

A. Subroutines READ and READIM 19 1/B11
B. Input Data Stream 20 1/B12

V. PROGRAM OUTPUTS 20 1/B12

VI. AUXILIARY PROGRAMS 33 1/C11

A. REDIM—The Redimension Program 33 1/C11
B. Support Programs for the DISCOS Flexible-Body Capability 42 1/D6

APPENDIX A—INPUT/OUTPUT SUBROUTINE EXPLANATIONS A-1 1/F12

APPENDIX B—SUPPORT PROGRAMS FOR THE DISCOS FLEXIBLE-BODY CAPABILITY B-1 2/A5

If the run number is equal to STOP (card columns 1 through 4), the run is terminated.

If the run number is not equal to STOP, the run continues in subroutine START as follows.

- Reads input card 2 for title card 1. Any keypunch symbols may be used in card columns 1 through 72.
- Reads input card 3 for title card 2. Any keypunch symbols may be used in card columns 1 through 72.
- Initializes page number as zero for use in subroutine PAGEHD.
- Interrogates computer for the date.

Run number, date, page number, user's name, title card 1, and title card 2 are transferred by a COMMON block labeled LSTART for use in other subroutines PAGEHD, PLOT1, PLOT2, PLOT3, and WTAPE.

Subroutine START is used to start each computer run in the FORMA system and will normally be the first subroutine called in a computer program. For example, pertinent statements from a program using START could be:

```
1 CALL START
      .
      .
      .
  GO TO 1
  END
```

WRITE

Subroutine WRITE writes a matrix of real numbers (a Fortran term for numbers with a decimal point) on paper. A group of up to ten consecutive elements from a row of the matrix are printed on each line. If all elements of a group are zero, printing of this line is suppressed.

Each matrix that is printed begins on a new page. Each page of printout contains the page heading given by subroutine PAGEHD, the name of the matrix, and the row size and column size of the matrix, followed by the matrix data. On any line of matrix data, the first integer number is the row number of the matrix elements on that line. The second integer number is the column number of the matrix element in the first data field. The next group of real numbers (up to ten) are the values of the matrix elements. This group of matrix elements is given in consecutive column order.

WRITIM

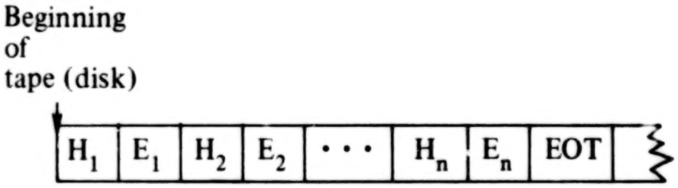
Subroutine WRITIM writes a matrix of integer numbers on paper. A group of up to 20 consecutive elements from a row of the matrix are printed on each line. If all elements of a group are zero, printing of this line is suppressed.

Each matrix that is printed beings on a new page. Each page of printout contains the page heading given by subroutine PAGEHD, the name of the matrix, and the row size and column size of the matrix, followed by the matrix data. On any line of matrix data, the first integer number is the row number of the matrix elements on that line. The second integer number is the column number of the matrix element in the first data field. The next group of integer numbers (up to 20) are the values of the matrix elements. This group of matrix elements is given in consecutive column order.

WTAPE

Subroutine WTAPE writes matrix data at the end of existing written matrix data on a FORMA tape. (A disk is preferred; see the following.) Each set of matrix data consists of two logical records. The first record contains the matrix heading (tape identification, location number, run number, matrix name, number of rows of matrix, number of columns of matrix, date, and the word "dense"). The second record consists of the matrix elements.

The following sketch is a schematic representation of the tape (disk):



where

H_i = Matrix heading of the i^{th} written matrix

E_i = Matrix elements of the i^{th} written matrix

EOT = End of tape--Data written by subroutine WTAPE or INTAPE that all FORMA tape subroutines recognize as the end of written data.

Each vertical line is an end of logical record put on by the computer system's routines. The tape is written in binary form as opposed to binary coded decimal (BCD) form.

To find the end of written matrix data, a search is made of the matrix headings until the EOT is found. For this reason, a "new" tape (disk) must be initialized with subroutine INTAPE so that the tape (disk) contains an EOT. A "new" tape (disk) is defined as a tape (disk) for which it is desired to start writing matrix data at the front of the tape (disk). Thus, a "new" tape (disk) could be one that contains obsolete FORMA matrix data, as well as one on which the FORMA system has never written. When the EOT is found, a backspace operation is performed over the EOT, and the current matrix heading, current matrix elements, and a new EOT are then written.

A disk is preferred to a tape for the following reason. Because the physical separation of the read and write heads on most tape drives may cause tolerance problems, backspacing over the EOT is usually not successful. Instead of ending up positioned in front of the EOT, the write head is often positioned in front of the previous matrix elements (E_n in the foregoing sketch). The current matrix heading will be written over the previous matrix elements. This causes problems later when an attempt is made to read the records written on the tape. To alleviate this problem, it is strongly recommended that an intermediate device such as a disk be used with all FORMA tape subroutines (INTAPE, LTAPE, RTAPE, WTAPE, and UPDATE). At the beginning of a computer run, the existing tape should be copied onto the disk by using computer control cards. Likewise, at the end of the run, the disk should be copied back onto tape by using computer control cards.

APPENDIX B

SUPPORT PROGRAMS FOR THE DISCOS FLEXIBLE-BODY CAPABILITY

```
C
C
C                                                        DEBUG *    1
C
C  PROGRAM MAIN--DYNAMIC SIMULATION OF CONTROLLED INTERCONNECTED SYSTEM
C     OF FLEXIBLE BODIES.  PROGRAMMED FOR GODDARD SPACE FLIGHT CENTER
C
C
C     CREDITS:
C              ALL THEORETICAL DEVELOPEMENT AND EQUATION CODEING HAS
C              BEEN DONE BY:
C                      CARL S. BODLEY      (MARTIN MARIETTA CORPORATION)
C                      A. DARRELL DEVERS   (MARTIN MARIETTA CORPORATION)
C                      A. COLTON PARK      (MARTIN MARIETTA CORPORATION)
C
C              BASIC PROBLEM DEFINITION, REQUIRED STABILITY ANALYSIS
C              CAPABILITY AND THE BASIC APPROACH TO BE USED FOR DATA
C              INPUT HAS BEEN PROVIDED BY:
C                      HAROLD P. FRISCH    (GODDARD SPACE FLIGHT CENTER)
C                      JAMES H. DONOHUE    (GODDARD SPACE FLIGHT CENTER)
C
C              THE PROGRAM HAS BEEN REVIEWED AND FURTHER ANNOTATED TO
C              CROSS REFERENCE WITH THE THEORETICAL MANUAL BY:
C                      HAROLD P. FRISCH    (GODDARD SPACE FLIGHT CENTER)
C
C              REQUIREMENTS FOR MAKING THE PROGRAM COMPATABLE WITH
C              THE GSFC IBM-360 SYSTEM WAS DEFINED BY:
C                      REGINALD MITCHELL   (GODDARD SPACE FLIGHT CENTER)
C
C              THE TECHNICAL MONITOR WAS:
C                      JOSEPH P. YOUNG     (GODDARD SPACE FLIGHT CENTER)
C
C
C
C
C     NOTE *****    AVOID USE OF QUAD PRECISION AT GSFC
C                   COMPUTATION SPEED AT LEAST 30 TIMES
C                   SLOWER THAN DOUBLE PRECISION
C
C                    DISCOS SUBROUTINE LOCATION
C
C       SUBROUTINE    NEW LOCATION   OLD LOCATION    OVERLAY  PRECISION
C                     DEBUG NUMBER   TAPE 11702                TAPE 11702
C
C          ADDT           105         9331 - 9341     ALPHA-2
C          ADD3            27         2291 - 2305     BETA -1
C          ADT            104         9319 - 9330     ALPHA-2
C          ALPHAA          95         8861 - 8881     ALPHA-4
C          ASIMCR          65         5819 - 5940     ALPHA-4
C          ASQR           113         NEW             ALPHA-4
C          EADT            96         8882 - 8918     ALPHA-4
C          EAKSLV          52         4380 - 4398     ALPHA-2
C          EDOTCP          51         4236 - 4379     ALPHA-2
C          EMGENR          45         3766 - 3841     ALPHA-2
C          ESGENR          47         3875 - 3924     ALPHA-2
C          ETABA           33         2745 - 2798     BETA -2
C          CANCOR          78         7368 - 7450     ALPHA-4
C          COMENT           7          560 -  590
C          CONTRL         107         9399 - 9575     ALPHA-2
C          CREA            21         2021 - 2046     BETA -1
```

```
C        CREADD      19        1859 - 1948   BETA -1            0   61
C        CREB        22        2049 - 2089   BETA -1            0   62
C        CREC        23        2090 - 2216   BETA -1            0   63
C        CREE        20        1949 - 2020   BETA -1            0   64
C        CREMCD      18        1791 - 1858   BETA -1            0   65
C        CRETS       17        1733 - 1790   BETA -1            0   66
C        DCCM2       42        3535 - 3557   ALPHA-2            0   67
C        DCGRRT      70        6523 - 6559   ALPHA-4            0   68
C        DEF 1      112        NEW (COMMENT CARDS, UNUSED)      0   69
C        DEF 2      113        NEW (COMMENT CARDS, UNUSED)      0   70
C        DEF 3      114        NEW (COMMENT CARDS, UNUSED)      0   71
C        DEF 4      115        NEW (COMMENT CARDS, UNUSED)      0   72
C        DEF 5      116        NEW COMMENT CARDS, UNUSED        0   73
C        DFGRMB      80        7479 - 7537   ALPHA-4            0   74
C        DIMAG       76        7354 - 7360                      0   75
C        DREAL       77        7361 - 7397                      0   76
C        DYNS10       2          61 -  358   ALPHA-1            0   77
C        DYNS20      37        2821 - 3005   ALPHA-2            0   78
C        DYNS30      62        5024 - 5096   ALPHA-3            0   79
C        DYNS40      64        5251 - 5818   ALPHA-4            0   80
C        EIGVEC     117        NEW           ALPHA-4            0   81
C        ENGMEM      58        4658 - 4742   ALPHA-2            0   82
C        EQADD      110        9633 - 9660   ALPHA-2            0   83
C        EXTCH      108        9574 - 9593   ALPHA-2            0   84
C        FETCH       25        2239 - 2272   BETA -1            0   85
C        FINDT       66        5541 - 6059   ALPHA-4            0   86
C        FINDC       33        3006 - 3155   ALPHA-2            0   87
C        FIT         75        7332 - 7353   ALPHA-4            0   88
C        FORMB       79        7451 - 7478   ALPHA-4            0   89
C        GAUSSI      31        2484 - 2547                      0   90
C        GETBMD      48        3925 - 4022   ALPHA-2            0   91
C        GMISC      111        9661 - 9669   ALPHA-2            0   92
C        GRVGRD      50        4141 - 4255   ALPHA-2            0   93
C        INTAPE       9         619 -  643                      0   94
C        INVINP      41        3492 - 3534                      0   95
C        KRINGE     106        9342 - 9398   ALPHA-2            0   96
C        LINEAR      40        3350 - 3491   ALPHA-2            0   97
C        LPLTWR     100        9159 - 9184   ALPHA-4            0   98
C        LPRNT      101        9185 - 9270   ALPHA-4            0   99
C        LTAPE       14         967 - 1029                      0  100
C        LTGRCL     103        9295 - 9318   ALPHA-4            0  101
C        LTRESP      99        8998 - 9158   ALPHA-4            0  102
C        MAIN         1           1 -   60                      0  103
C        MGEN        49        4023 - 4140   ALPHA-2            0  104
C        MRIGID      28        2306 - 2409   ALPHA-1            0  105
C        MSMCDC      15        1030 - 1461   BETA -2            0  106
C        MSMCDL      16        1462 - 1732   BETA -1            0  107
C        MULTA       34        2704 - 2744                      0  108
C        MULTAD      55        4431 - 4445                      0  109
C        MULTB       97        8919 - 8952   ALPHA-4            0  110
C        MULTSR      56        4446 - 4463   ALPHA-2            0  111
C        MULT3       54        4416 - 4430                      0  112
C        NXPLCT      90        8386 - 8516   ALPHA-4            0  113
C        NUMS        68        6325 - 6435   ALPHA-4            0  114
C        NYPLCT      91        8517 - 8629   ALPHA-4            0  115
C        PAGEHD       8         591 -  618                      0  116
C        PLOTSS      94        8801 - 8860   ALPHA-4            0  117
C        PLOTWR      61        4977 - 5023   ALPHA-2            0  118
C        PLTCAR      63        5097 - 5250   ALPHA-3            0  119
C        PRNTGD      60        4854 - 4976   ALPHA-2            0  120
```

```
C             FRO          56          2799 - 2520     BETA -2                    0
C             CHECK        65          7651 - 7696     ALPHA-4   REAL*16          0
C             CHORWR       64          7797 - 7550     ALPHA-4   REAL*16          0
C             CH2          67          8032 - 8113     ALPHA-4   REAL*16          0
C             FLAG         10           644 -  794                                0
C             FLAGIM        3           369 -  433                                0
C             FLVADD       98          8553 - 8997     ALPHA-4                    0
C             FLVISE       30          2428 - 2463     BETA -2                    0
C             FRALAM       59          4743 - 4833     ALPHA-2                    0
C             FLOCGS       74          7066 - 7331     ALPHA-4                    0
C             FLPLOT       92          8630 - 8751     ALPHA-4                    0
C             FMVZRO       71          6570 - 6599     ALPHA-4                    0
C             FLTOF        44          3583 - 3765     ALPHA-2                    0
C             FLTOS        46          3842 - 3874     ALPHA-2                    0
C             FLTTR        32          2584 - 2685                                0
C             FTAPE        11           795 -  849                                0
C             FTOP         81          7508 - 7579     ALPHA-4                    0
C             FWRITE       85          8114 - 8160     ALPHA-4                    0
C             SATO         26          2273 - 2290     BETA -1                    0
C             SCALE        95          8762 - 8800     ALPHA-4                    0
C             SFREG2       73          6610 - 7065     ALPHA-4                    0
C             SHAFTT      109          9604 - 9632     ALPHA-2                    0
C             SIFT         72          6591 - 6639     ALPHA-4                    0
C             SKEWVS       53          4399 - 4413                                0
C             SPLOT        89          8161 - 8335     ALPHA-4                    0
C             START         6           511 -  559                                0
C             STORE        24          2219 - 2238     BETA -1                    0
C             SUDDIA       85          7589 - 8031     ALPHA-4   REAL*16          0
C             TFPLY        69          6456 - 6522                                0
C             TFTYPE       67          6070 - 6324                                0
C             TORQUE       57          4464 - 4657     ALPHA-2                    0
C             TRFB         82          7680 - 7743     ALPHA-4                    0
C             TTFF         83          7744 - 7796     ALPHA-4                    0
C             UNITY        33          2686 - 2703                                0
C             WRITE        12           850 -  908                                0
C             WRITES       43          3558 - 3582                                0
C             WRITIM        4           434 -  483                                0
C             WRITIS        5           484 -  510                                0
C             WTAPE        13           909 -  950                                0
C             YOUT         39          3156 - 3349     ALPHA-2                    0
C             YOUTL       102          9271 - 9294     ALPHA-4                    0
C             ZERO         29          2410 - 2427                                0
C                                                                                 0
C                                                                                 0
        IMPLICIT REAL*8(A-H,O-Z)                                                  0
C                                                                                 0
              COMMON /DRATIO/                                                     0
     *      IFL1,IFL2,DRVEC(150)                                                100
              COMMON /GODATA/                                                     0
     *      GAMGI(3),GMAG,RCMAG                                                   0
              COMMON /ILINER/                                                     0
     *      IFLNER                                                                0
              COMMON /MAXMUM/                                                     0
     *      N3MAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMOBJD,KMU,KY,KU                     0
              COMMON /MISCNO/                                                     0
     *      NOPRNT,NOPLOT                                                         0
              COMMON /NUMBRS/                                                     0
     *      ZRO,ONE,TWO,TRES                                                      0
              COMMON /PLTDTA/                                                     0
     *      NRPLOT,NCPLOT                                                         0
```

```
             COMMON /SPECIF/                                              0
     *      BETAH(6, 6),BETAHD(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),    16
     *      DH(3,30),DS(3,30),IMO(3, 5),NMOW(6, 5),IFISAW(15),           17
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18
     *      LOCC(14),LENO(14),NU,NBETA,NLAM,NEQ                          19
             COMMON /TAPENO/                                              0
     *      NTAPE1,NTAPE2,NTAPE3                                          0
C                                                                         0
      IFL1 = 0                                                            0
C                                                                         0
      NTAPE1 = 1                                                          0
      NTAPE2 = 2                                                          0
      NTAPE3 = 3                                                          0
C        DEFINE MAX SIZES OF ARRAYS. THESE MAY ONLY BE CHANGED VIA        0
C                  THE REDIMENSION PROGRAM                                0
      NBMAX  =  6                                                        41
      NHMAX  =  6                                                        42
      NSPMAX = 15                                                        43
      NMWMAX =  5                                                        44
      NMWBCD =  4                                                        45
      NMCBCD = 12                                                        46
      KMU    = 22                                                        47
      KY     = 250                                                       48
      KU     = 113                                                       49
C                                                                         0
      ZRO  = 0.D 0                                                        0
      ONE  = 1.D 0                                                        0
      TWO  = 2.D 0                                                        0
      THRES = 3.D 0                                                       0
C                                                                         0
  995 CALL START                                                          0
      CALL COMENT                                                         0
C                                                                         0
      REWIND NTAPE1                                                       0
      REWIND NTAPE2                                                       0
      REWIND NTAPE3                                                       0
C                                                                         0
C                       CALL DYNS10 TO READ IN AND PROCESS THE DATA WHICH 0
C                       DEFINES THE COUPLED BODY SIMULATION MODEL         0
      CALL DYNS10                                                         0
C                                                                         0
C                       CALL DYNS20 TO DERIVE AND INTEGRATE THE NON-LINEAR 0
C                       COUPLED BODY EQUATIONS OF MOTION OR CALL THE ROUTINES 0
C                       WHICH WILL NUMERICALLY LINEARIZE THE EQUATIONS    0
      CALL DYNS20                                                         0
C                                                                         0
C                       CALL DYNS40 TO GO THROUGH ALL LINEAR STABILITY    0
C                       ANALYSIS OF THE COUPLED EQUATIONS OF MOTION       0
      IF (IFLNER .EQ. 1) CALL DYNS40                                      0
C                                                                         0
C                       CALL DYNS30 ONLY IF PLOTTING IS REQUESTED         0
      IF (NOPLOT .GT. 0) CALL DYNS30                                      0
C                                                                         0
      GO TO 995                                                           0
C                                                                         0
      END                                                                 0
      SUBROUTINE DYNSIO                                                 0
C                                                      DEBUG #    2     0
      IMPLICIT REAL*8(A-H,O-Z)                                          0
C                                                                       0
C                                                                       C
            COMMON /CONPAR/                                             0
     *      CNTDTA(150)                                                95
            COMMON /GGDATA/                                             0
     *      GAMGI(3),GMAG,RCMAG                                         0
            COMMON /ILINER/                                             0
     *      IFLNER                                                      C
            COMMON /MAXNUM/                                             0
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMODJJ,K4J,KY,KU           0
            COMMON /MISCNO/                                             0
     *      NOPRNT,NOPLOT                                               C
            COMMON /NHNS  /                                             0
     *      NHPOI( 6), NSPOI( 6)                                       12
            COMMON /SPECIF/                                             0
     *      BETAH(6, 6),BETAHD(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),  16
     *      DH(3,35),DS(3,30),IMO(3, 5),NMOW(6, 6),IFISMW(15),         17
     *      NB,NH,NSPT,NGIMG,NDELTA,ITOPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18
     *      LOCC(14),LENG(14),NQ,NBETA,NLAM,NEQ                        19
            COMMON /SUMMRY/                                             0
     *      ASUMRY(15,6),ISUMRY(15,3),KSUMRY                           96
            COMMON /TAPENO/                                             0
     *      NTAPE1,NTAPE2,NTAPE3                                        C
            COMMON /TIMESS/                                             0
     *      STARTT,DELTAT,T,ENDT,TMST                                   0
C                                                                       0
      DIMENSION WV(5),TMDATA(3),IPDATA(3)                               C
      EQUIVALENCE (GAMGI(1),WV(1))                                      0
C                                                                       0
C                                                                       0
C     NB     = NO. OF BODIES                                            0
C     NH     = NO. OF HINGES    (.GE. NB)                               0
C     IRGFLX(L) = (0 IF L IS RIGID), (M(L) IF L IS FLEXIBLE)            0
C                     M(L) = NO. OF MODES OF BODY (L)                   C
C                     L=1,NB                                            C
C     NMOW(L) = NO. MOMENTUM WHEELS/BODY(L), L=1,NB                     0
C     ITOPOL(1,1) = 1    (BODY 1)                                       0
C     ITOPOL(2,1) = 0    (INERTIAL REFERENCE)                           0
C     ITOPOL(1,K) = L1,   (K .GT. 1)                                    0
C     ITOPOL(2,K) = L2,   (K .GT. 1)                                    C
C             BODY(L1) REL. TO BODY(L2),  K=1,NH.                       0
C     IHDATA(1,K) = ITYPE (EULER PERMUTATION 1,2...12), K=1,NH          0
C           (J,K) = 0  (FORCED/FREE)                                    C
C           (J,K) = 1  (FIXED CONSTRAINT)                               0
C           (J,K) = 2  (RHEONOMIC CONSTRAINT),   --  J=2,7,   K=1,NH    C
C     BETAH(J,K) = INITIAL HINGE COORDINATES, (J=1,6,  K=1,NH)          0
C     BETAHD(J,K) = INITIAL HINGE RATES,      (J=1,6,  K=1,NH)          C
C     AM(L)  = MASS OF BODY(L),  L=1,NB                                 0
C     SMOM(I,L) = STATIC MOMENT OF BODY(L),  L=1,NB                     0
C     AIN(I,L)  = MOMENT OF INERTIA PROPERTIES OF BODY(L), I=1,6,  L=1,NB  C
C                                                                       C
C                                                                       C
 1001 FORMAT (16I5)                                                     0
 2001 FORMAT (//15X46HSUMMARY OF DYNAMIC-SIMULATION-PROGRAM INPUT DATA, 0
     *  10(2H *),//3X12HACTUAL SIZES5X13HMAXIMUM SIZES4X12HINTEGRATION  0
     *  4HDATA 12X21HGRAVITY GRADIENT DATA17X10HMISC. DATA,             0
     *  /3X13(1H-),4X13(1H-),4X16(1H-),5X37(1H-),6X11(1H-),             0
     * /3X9HNB      = I4,4X9HNBMAX  = I4,4X9HSTARTT = 1PD10.3,4X7HG1   =
     * 1PD10.3,4X8HGAMA1 = 1PD10.3,4X9HNOPRNT = I2,
     * /3X9HNH      = I4,4X9HNHMAX  = I4,4X9HDELTAT = 1PD10.3,4X7HG2   =
     * 1PD10.3,4X8HGAMA2 = 1PD10.3,4X9HNOPLOT = I2,
     * /3X9HNSPT    = I4,4X9HNSPMAX = I4,4X9HENDT   = 1PD10.3,4X7HG3   =
     * 1PD10.3,4X8HGAMA3 = 1PD10.3,4X9HIFLNER = I2,
     * /3X9HNOFMC   = I4,4X9HNMWMAX = I4,                27X7HGMAG =
     * 1PD10.3,4X8HRCMAG = 1PD10.3,
     * /3X9HNDELTA = I4,4X9HNMWBOD = I4,
     * /3X9HNC      = I4,4X9HNMODCD = I4,
     * /3X9HNBETA   = I4,4X9HKMU    = I4,
     * /3X9HNLAM    = I4,4X9HKY     = I4,
     * /3X9HNEQ     = I4,4X9HKG     = I4)
 2002 FORMAT (//1X49HTHE TOPOLOGY ARRAY (ITOPOL) FOR THIS CASE FOLLOWS)
 2003 FORMAT (//1X50HTHE CONSTRAINT SPECIFICATIONS FOR THIS CASE FOLLOW)
 2004 FORMAT (//1X39HTHE SPECIFIED INITIAL HINGE ANGLES AND
     *     28HDISPLACEMENTS (BETAH) FOLLOW)
 2005 FORMAT (//1X49HTHE SPECIFIED INITIAL HINGE RATES (BETAHD) FOLLOW)
 2006 FORMAT (//1X45HTHE NO. OF ELASTIC MODES/BODY ARRAY (IRGFLX)
     * 7HFOLLOWS)
 2007 FORMAT (//1X47HTHE NO. OF P/Q HINGE POINTS/BODY ARRAY (NHPOI)
     * 7HFOLLOWS)
 2008 FORMAT (//1X44HTHE NO. OF SENSOR POINTS/BODY ARRAY (NSPOI)
     * 7HFOLLOWS)
 2009 FORMAT (//1X40HTHE MOM. WHEEL/BODY TABLE (NMGW) FOLLOWS )
 2010 FORMAT (//1X44HTHE STATE VECTOR LENGTH ARRAY (LENJ) FOLLOWS )
 2011 FORMAT (//1X46HTHE STATE VECTOR LOCATION ARRAY (LOCO) FOLLOWS)
 2012 FORMAT (//1X50HTHE SPECIFIED SENSOR POINT/BODY CORRELATION ARRAY
     * 15H(IFTSMW) FOLLOWS )
 2013 FORMAT (//1X43HTHE FOLLOWING DATA IS SPECIFIED MOM. WHEEL
     * 47HINFORMATION (IF ANY) AND CONTROLLER INFORMATION /1X90(1H-))
 2014 FORMAT (//1X45HTHE SPECIFIED MOM. WHEEL CONTROL ARRAY (IMC)
     * 7HFOLLOWS)
 2015 FORMAT (//1X50HTHE SPECIFIED MOM. WHEEL RATES AND INERTIAS (AMO)
     * 7HFOLLOW  )
 2016 FORMAT (//1X,43HTHE SPECIFIED CONTROLLER INITIAL CONDITIONS AND
     * 22HCHARACTERISTICS FOLLOW  /3X30H(THE FIRST NDELTA ARE INITIAL
     *  38HCONTROLLER STATE VARIABLES, THERE ARE I3,12H ADDITIONAL
     *  19HCONTROL PARAMETERS))
 2110 FORMAT (1H1, 28HINPUT DATA ERROR, NERROR =  I3)
C
      DATA NIT,NOT / 5, 6/
C
CCCCC
      KCONT = 150
      KSUMRY = 15
CCCCC
C
C******************** READ BASIC SYSTEM SIZES
      READ (NIT,1001) NB,NH,NSPT,NOFMC,NDELTA
C
                                                      NERROR =  1
C******************** READ SYSTEM TOPOLOGY
      CALL READIM (ITOPOL,N1,N2,2,NHMAX)
      IF (N1.NE.2 .OR. N2.NE.NH .OR. NH.LT.NB) GO TO 999
                                                      NERROR =  2
C******************** READ BODY TYPE AND NUMBER OF MODES IF FLEXIBLE
      CALL READIM (IRGFLX,N1,N2,1,NBMAX)
      IF (N2 .NE. NB) GO TO 999
                                                      NERROR =  3
```

```
C****************** READ SENSOR POINT LOCATION INFORMATION
      CALL READI4 (IFISMA,N1,N2,1,NSPMAX)
      IF (N2 .NE. NSPT) GO TO 999
                                                       NERROR =  4
C****************** READ CONSTRAINT DATA FOR EACH HINGE POINT
      CALL READI4 (IHDATA,N1,N2, 7,NHMAX)
      IF (N1.NE.7 .OR. N2.NE.NH) GO TO 999
                                                       NERROR =  5
C****************** READ CONTIGUOUS HINGE FRAME ORIENTATION DATA
      CALL READ  (BETAH,N1,N2, 6,NHMAX)
      IF (N1.NE.6 .OR. N2.NE.NH) GO TO 999
                                                       NERROR =  6
C****************** READ CONTIGUOUS HINGE FRAME RATE DATA
      CALL READ  (BETAHD,N1,N2, 6,NHMAX)
      IF (N1.NE.6 .OR. N2.NE.NH) GO TO 999
                                                       NERROR =  7
      IF (ITOPOL(1,1).NE.1 .OR. ITOPOL(2,1).NE.0) GO TO 999
                                                       NERROR =  8
      DO 605 J=2,NH
      IF (ITOPOL(1,J) .EQ. ITOPOL(2,J)) GO TO 999
  605 CONTINUE
C
CCC   PRELIMINARY TOPOLOGY CHECK, COMPLETE CHECK DONE BY ROTDM ....
                                                       NERROR =  9
      DO 610 N=1,NB
      DO 610 I=1,2
      DO 610 J=2,NH
      IF (ITOPOL(I,J) .EQ. N) GO TO 612
  610 CONTINUE
      GO TO 999
  612 CONTINUE
                                                       NERROR = 10
      DO 620 J=1,NH
      IF (IHFLEX(J).LT.0 .OR. IHFLEX(J).GT.NMODJJ) GO TO 999
  620 CONTINUE
                                                       NERROR = 11
      DO 625 J=1,NSPT
      IF (IFISMA(J).LT.1 .OR. IFISMA(J).GT.NB) GO TO 999
  625 CONTINUE
                                                       NERROR = 12
      DO 630 J=1,NH
      IF (IHDATA(1,J).LT.1 .OR. IHDATA(1,J).GT.12) GO TO 999
      DO 630 I=2,7
      IF (IHDATA(I,J).LT.0 .OR. IHDATA(I,J).GT.2) GO TO 999
  630 CONTINUE
C
C                                NHPOI(N) = NUMBER OF HINGES ON BODY N
C                                NSPOI(N) = NUMBER OF SENSORS ON BODY N
C
      DO 5 N=1,NB
      NHPOI(N) = 0
      NSPOI(N) = 0
      DO 10 I=1,2
      DO 10 J=2,NH
      IF (ITOPOL(I,J) .EQ. N) NHPOI(N) = NHPOI(N) + 1
   10 CONTINUE
      DO 15 J=1,NSPT
      IF (IFISMA(J) .EQ. N) NSPOI(N) = NSPOI(N) + 1
   15 CONTINUE
    5 CONTINUE
```

```
C
      I1 = NMWBOD + 2
      DO 40 I=1,I1
      DO 40 J=1,NBMAX
   40 NMOW(I,J) = 0
      IF (NOFMO .EQ. 0) GO TO 41
                                                               NERROR = 13
C******************** READ BASIC MOMENTUM WHEEL DATA
      CALL READIM (IMO,N1,N2, 3,NMWMAX)
      IF (N1.NE.3 .OR. N2.NE.NOFMO) GO TO 999
                                                               NERROR = 14
C******************** READ MOMENTUM WHEEL RATE + INERTIA
      CALL READ   (AMO,N1,N2, 2,NMWMAX)
      IF (N1.NE.2 .OR. N2.NE.NOFMO) GO TO 999
                                                               NERROR = 15
      DO 635 J=1,NOFMO
      IF (IMO(1,J).LT.1 .OR. IMO(1,J).GT.NSPT) GO TO 999
      IF (IMO(2,J).LT.1 .OR. IMO(2,J).GT.3) GO TO 999
      IF (IMO(3,J).LT.0 .OR. IMO(3,J).GT.1) GO TO 999
      IF (AMO(2,J) .LE. 0.0 0) GO TO 999
  635 CONTINUE
                                                               NERROR = 16
      IF (NOFMO .EQ. 1) GO TO 641
      N1 = NOFMO - 1
      DO 640 I=1,N1
      IP1 = I + 1
      DO 640 J=IP1,NOFMO
      IF (IMO(1,I) .EQ. IMO(1,J)) GO TO 999
  640 CONTINUE
                                                               NERROR = 17
C                             NMOW = MOMENTUM WHEEL/BODY TABLE
C
  641 DO 45 N=1,NOFMO
      NPTS = IMO(1,N)
      NBOD = IPTSAW(NPTS)
      NMOW(1,NBOD) = NMOW(1,NBOD) + 1
      IF (IMO(3,N) .NE. 0) NMOW(2,NBOD) = NMOW(2,NBOD) + 1
      IC = NMOW(1,NBOD) + 2
      IF (IC .GT. I1) GO TO 999
   45 NMOW(IC,NBOD) = N
   41 CONTINUE
C
C                             LENU = STATE VECTOR LENGTH ARRAY
C                             LOCU = STATE VECTOR LOCATION ARRAY
C
      NEC = 0
      DO 50 N=1,NB
      NEC = NEC + IRGFLX(N)
   50 LENU(N) = 6 + IRGFLX(N) + NMOW(2,N)
      LOCU(1) = 1
      DO 55 N=2,NB
   55 LOCU(N) = LOCU(N - 1) + LENU(N - 1)
      NU = LOCU(NB) + LENU(NB) - 1
      DO 56 N=1,NB
   56 LENU(N+NB) = IRGFLX(N)
      DO 57 N=1,NB
      NM1 = N - 1
   57 LOCU(N+NB) = LOCU(NM1+NB) + LENU(NM1+NB)
      NBETA = 0
      NLAM = 0
      DO 60 I=2,7
      DO 60 J=1,NH
      IF (IHDATA(I,J) .NE. 1) NBETA = NBETA + 1
      IF (IHDATA(I,J) .NE. 0) NLAM = NLAM + 1
   60 CONTINUE
      NEQ = NEQ + NBETA + NDELTA + NU
      LENO(2*NB+1) = NBETA
      LENO(2*NB+2) = NDELTA
      LOCO(2*NB+1) = LOCO(2*NB) + LENO(2*NB)
      LOCO(2*NB+2) = LOCO(2*NB+1) + NBETA
C
C******************** READ TIME HISTORY INTEGRATION CONTROL
      CALL READ (TMDATA,N1,N2,1,3)
C
C******************** READ PRINT, PLOT, LINEAR, NON-LINEAR FLAGS
      CALL READIM (IPDATA,N1,N2,1,3)
C
C******************** READ CONTROL SYSTEM DATA
      CALL READ (CNTDTA,N1,NCNPAR,1,KCONT)
                                                    NERROR = 18
C******************** READ GRAVITY GRADIENT DATA
      CALL READ  (WV,N1,N2,1,5)
C
      IF (N2 .NE. 4) GO TO 999
      RCMAG = WV(4)
      GMAG = DSQRT(GAMGI(1)**2 + GAMGI(2)**2 + GAMGI(3)**2)
CC WV(5) IS NOW RCMAG, WV(4) IS GMAG
      IF (GMAG .EQ. 0.D 0) GO TO 75
      IF (RCMAG .LE. 1.D 0) GO TO 999
C         COMPUTE UNIT VECTOR, INERTIAL COORDINATES IN DIRECTION
C         OF GRAVITY FIELD
      DO 70 J=1,3
   70 GAMGI(J) = GAMGI(J)/GMAG
C
   75 CALL PAGEHD
      STARTT = TMDATA(1)
      LELO = 2*NB + 2
      DELTAT = TMDATA(2)
      ENDT = TMDATA(3)
      NOPRNT = IPDATA(1)
      NOPLOT = IPDATA(2)
      IFLNER = IPDATA(3)
      G1 = GMAG*GAMGI(1)
      G2 = GMAG*GAMGI(2)
      G3 = GMAG*GAMGI(3)
      WRITE (NOT,2001) NB,NBMAX,STARTT,G1,GAMGI(1),NOPRNT,NH,NHMAX,
     *  DELTAT,G2,GAMGI(2),NOPLOT,NGPT,NGPMAX,ENDT,G3,GAMGI(3),IFLNER,
     *  NOFMG,NMWMAX,GMAG,RCMAG,NDELTA,NMWBOD,NU,NMDBOD,NBETA,KMU,
     *  NLAM,KY,NEQ,KU
      WRITE (NOT,2002)
      CALL WRITIS (ITOPOL,2,NH,2)
      WRITE (NOT,2003)
      CALL WRITIS (IHDATA,7,NH,7)
      WRITE (NOT,2004)
      CALL WRITES (BETAH,6,NH,6)
      WRITE (NOT,2005)
      CALL WRITES (BETAHD,6,NH,6)
      CALL PAGEHD
      WRITE (NOT,2006)
      CALL WRITIS (IRGFLX,1,NB,1)
      WRITE (NOT,2007)
      CALL WRITIS (NHPOI,1,NB,1)
      WRITE (NOT,2008)
      CALL WRITIS (NSPOI,1,NB,1)
      WRITE (NOT,2009)
      CALL WRITIS (NMOW,11,NB,11)
      WRITE (NOT,2010)
      CALL WRITIS (LENO,1,LELO,1)
      WRITE (NOT,2011)
      CALL WRITIS (LOCO,1,LELO,1)
      WRITE (NOT,2012)
      CALL WRITIS (IFTSMW,1,NSPT,1)
      CALL PAGEHD
      WRITE (NOT,2013)
      IF (NOFMO .NE. 0) WRITE (NOT,2014)
      IF (NOFMO .NE. 0) CALL WRITIS (IMO,3,NOFMO,3)
      IF (NOFMO .NE. 0) WRITE (NOT,2015)
      IF (NOFMO .NE. 0) CALL WRITES (AMO,2,NOFMO,2)
      NCNTRL = NCNPAR - NDELTA
      WRITE (NOT,2016) NCNTRL
      CALL WRITES (CNTDTA,1,NCNPAR,1)
C
C     GET BODY DATA FOR EACH OF THE NB BODIES
      DO 20 N=1,NB
      IF (IRGFLX(N) .EQ. 0) GO TO 25
C
C******************** READ FLAG TO SPECIFY LUMPED OR CONSISTANT MASS
C******************** FLEXIBLE BODY DATA
      READ (NIT,1001) NTYPE
C
C               MSMODL PROCESSES LUMPED MASS DATA FOR FLEXIBLE BODIES
      IF (NTYPE .EQ. 1) CALL MSMODL (N)
C
C               MSMODC PROCESSES CONSISTANT MASS DATA FOR FLEXIBLE BODIES
      IF (NTYPE .EQ. 2) CALL MSMODC (N)
C
      GO TO 20
C               MRIGID PROCESSES DATA FOR RIGID BODIES
   25 CALL MRIGID (N)
C
   20 CONTINUE
C
      RETURN
  999 WRITE (NOT,2110) NERROR
      STOP
      END
      SUBROUTINE READIM (IA,NR,NC,KR,KC)
C                                                        DEBUG #    3

      REAL*8 ANAME,REMRK
      DIMENSION IA(KR,1),IX(14),REMRK(3)
      DATA NIT,NOT,IASTRS/ 5, 6, 1H* /
C
CC--TAKEN FROM FORMA.
CC--ACKNOWLEDGEMENT GIVEN TO RF HRUDA.
C
 1001 FORMAT (A6,I4,I5,9A6, 2XA1)
 1002 FORMAT (16I5)
 2001 FORMAT (//27H CARD INPUT INTEGER MATRIX A6, 2X 1H( I4,2H X I4,2H )
     *           2X 9A6,//)
 2002 FORMAT (//27H CARD INPUT INTEGER MATRIX A6, 2X 1H( I4,2H X I4,2H )
     *           3X 9HCONTINUED //)
 2004 FORMAT (1X 16I5)
 2005 FORMAT (15H0END OF READIM.)
 2006 FORMAT (1H1,37HERROR IN SUBROUTINE READIM, NERROR = ,I3)
C
C  READ IN HEADER CARD.
      READ (NIT,1001) ANAME,N1,N2,REMRK,IZ1
      IPRIN = 0
      IF (IZ1 .EQ. IASTRS) IPRIN = 1
      IF (IPRIN .EQ. 1) CALL PAGEHD
C
C  CARD READING SECTION.
      NR = N1
      NC = N2
      IF (IPRIN .EQ. 1) WRITE (NOT,2001) ANAME,NR,NC,REMRK
      NERROR = 1
      IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999
      NLINE = 0
      DO 105 I=1,NR
      DO 105 J=1,NC
  105 IA(I,J) = 0
  110 READ (NIT,1002) I,JS,IX
      IF (I.EQ.0 .AND. JS.EQ.0) GO TO 400
      NERROR = 2
      IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998
      JE = JS + 13
      IF (JE .LE. NC) GO TO 115
      JX = NC - JS + 2
      NERROR = 3
      DO 112 J=JX,14
      IF (IX(J) .NE. 0) GO TO 998
  112 CONTINUE
      JE = NC
  115 N = 0
      DO 120 J=JS,JE
      N = N + 1
  120 IA(I,J) = IX(N)
      NLINE = NLINE + 1
      IF (NLINE .LE. 47) GO TO 125
      IF (IPRIN .EQ. 1) CALL PAGEHD
      IF (IPRIN .EQ. 1) WRITE (NOT,2002) ANAME,NR,NC
      NLINE = 1
  125 IF (IPRIN .EQ. 1) WRITE (NOT,2004) I,JS,(IA(I,J),J=JS,JE)
      GO TO 110
C
  400 IF (IPRIN .EQ. 1) WRITE (NOT,2005)
      RETURN
C
```

```
  998 WRITE (NOT,2004) I,JS,IX
  999 WRITE (NOT,2006) NERROR
      STOP
      END
```

```
      SUBROUTINE WRITIM (IA,NR,NC,ANAME,KR)
C                                                            DEBUG #    4
      REAL*8 ANAME
      DIMENSION IA(KR,1)
      DATA NOT / 6/
C
CC-- TAKEN FROM FORMA --
CC -- ACKNOWLEDGEMENT GIVEN TO R L WOHLEN.
C
 2010 FORMAT (//15H OUTPUT MATRIX A8,2X 1H(I4,2H X I4,2H ) //
     *         16X,20(1X,1H( I2,1H))/)
 2020 FORMAT (//15H OUTPUT MATRIX A8,2X 1H(I4,2H X I4,2H )
     *          3X, 9HCONTINUED //16X,20(1X,1H( I2,1H))/)
 2030 FORMAT (1X,2I5,5X,20I5)
 2040 FORMAT (15H0END OF WRITIM.)
C
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.
      CALL PAGEHD
      WRITE (NOT,2010) ANAME,NR,NC,(L,L=1,20)
      NLINE = 0
C
      DO 60 I=1,NR
      NZERO = 0
      JS = 1
   10 JE = JS + 19
      IF (JE .GT. NC) JE = NC
C  SEE IF ELEMENTS ARE ZERO.
      DO 20 J=JS,JE
      IF (IA(I,J) .NE. 0) GO TO 30
   20 CONTINUE
      GO TO 40
   30 NLINE = NLINE + 1
      IF (NLINE .LE. 44) GO TO 35
      CALL PAGEHD
      WRITE (NOT,2020) ANAME,NR,NC,(L,L=1,20)
      NLINE = 1
   35 WRITE (NOT,2030) I,JS,(IA(I,J), J=JS,JE)
      NZERO = 1
   40 IF (JE .EQ. NC) GO TO 50
      JS = JS + 20
      GO TO 10
C  SKIP A SPACE BETWEEN EACH ROW IF  THERE ARE MORE THAN 20 COLUMNS
C  AND SOMETHING HAS BEEN WRITTEN.
   50 IF (NC.LE.20 .OR. NZERO.EQ. 0 .OR. I.EQ.NR)  GO TO 60
      NLING = NLINE + 1
      WRITE (NOT,2030)
   60 CONTINUE
C
      WRITE (NOT,2040)
      RETURN
      END
      SUBROUTINE WRITIS (IM,NR,NC,KR)
C                                                    DEBUG #    5
C     PRINT INTEGER TWO DIMENSIONAL ARRAYS IN FORMA FORMAT
      DIMENSION IM(KR,1), ICH(20)
      DATA NCT / 6/
      DATA ICH /
     *  4H( 1),4H( 2),4H( 3),4H( 4),4H( 5),4H( 6),4H( 7),4H( 8),
     *     4H( 9),4H(10),
     *  4H(11),4H(12),4H(13),4H(14),4H(15),4H(16),4H(17),4H(18),
     *     4H(19),4H(20)  /
C
 2001 FORMAT (17X20(1X,A4))
 2002 FORMAT (1X,2I5,5X,20I5)
C
      LR = 20
      IF (NC .LT. LR) LR = NC
      WRITE (NCT,2001) (ICH(L),L=1,LR)
      DO 60 I=1,NR
      JS = 1
   10 JE = JS + 19
      IF (JE .GT. NC) JE = NC
      WRITE (NCT,2002) I,JS, (IM(I,J),J=JS,JE)
      IF (JE .EQ. NC) GO TO 60
      JS = JS + 20
      GO TO 10
   60 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE START
C                                                    DEBUG #    6
      REAL*8 IRUNNO,IDATE,UNAME,TITLE1,TITLE2,STOP
      COMMON /LSTART/IRUNNO,IDATE,NPAGE
      COMMON /LSTRT1/ UNAME(3),TITLE1(12),TITLE2(12)
      COMMON /LZMODE/ AMODE(200)
      DATA NIT,NCT,STOP / 5, 6, 4HSTOP/
      DATA IIST/ 0 /
      LOGICAL DEBUG
      REAL*8 RDBUG
      COMMON /LDEBUG/ DEBUG(120)
      DATA RDBUG/6HDEBUG /
C
CCC  --- TAKEN FROM FORMA, ACKNOWLEDGEMENT GIVEN TO R. HRUDA, R. WOHLEN
C  READS INPUT CARD 1 FOR IRUNNO, UNAME.
C  CHECKS IRUNNO FOR STOP (I.E. IF IRUNNO = STOP, PROGRAM WILL
C  BE STOPPED). YOU SHOULD HAVE A STOP CARD (THE WORD STOP PUNCHED
C  STARTING IN COLUMN 1) AFTER YOUR REGULAR DATA DECK.
```

```
C  IF IRUNNO IS NOT EQUAL TO STOP, THE SUBROUTINE CONTINUES AS FOLLOWS.
C  READS INPUT CARD 2 FOR TITLE1.
C  READS INPUT CARD 3 FOR TITLE2.
C  SETS NPAGE = 0.
C  INTERCGATES COMPUTER TO DEFINE DATE AS AN A6.
C
C      INPUT ORDER
C  IRUNNO,UNAME       FORMAT (A6, 4X 3A6)
C  TITLE1             FORMAT (12A6)
C  TITLE2             FORMAT (12A6)
C
C      DEFINITIONS
C  IRUNNO = RUN NUMBER. (A6 FORMAT)
C  DATE   = DATE. (A8 FORMAT).
C  NPAGE  = PAGE NUMBER.
C  UNAME  = USERS NAME. (3A6 FORMAT)
C  TITLE1 = FIRST  TITLE. (12A6 FORMAT)
C  TITLE2 = SECOND TITLE. (12A6 FORMAT)
C
C          IF IRUNNO = 'DEBUG' THE DEBUG PRINTS CALLED
C          DEBUG(K) = .FALSE. SKIP DEBUG PRINT IN SUBROUTINE K
C                     .TRUE. PRINT
C
 1001 FORMAT (A6, 4X 3A6)
 1002 FORMAT (12A6)
 1003 FORMAT (60L1)
 2003 FORMAT (36H1END OF INPUT DATA HAS BEEN REACHED.)
C
      IF (IIST .EQ. 0) CALL MODESG (AMODE,0)
      IIST = 1
      READ (NIT,1001) IRUNNO,UNAME
      DO 12 I=1,120
   12 DEBUG(I) = .FALSE.
      IF(IRUNNO.NE.PDBUG) GO TO 11
      READ (NIT,1003) (DEBUG(I),I=1,120)
   11 CONTINUE
      IF (IRUNNO .NE. STOP) GO TO 10
      CALL EXITG (AMODE)
      WRITE (NOT,2003)
      STOP
C
   10 READ (NIT,1002) TITLE1
      READ (NIT,1002) TITLE2
      NPAGE = 0
      CALL DATE (IDATE)
      RETURN
      END
```

```
      SUBROUTINE COMENT
C                                                             DEBUG #    7
      REAL*8 IREMRK,IPGHD,P,QUIT
      DIMENSION  IREMRK(13)
      DATA NIT,NOT,QUIT,P / 5, 6, 6H000000, 1HP/
C
C  READ COMMENT CARDS AND PRINT THEM UNDER PAGE HEADING OF FORMA
```

```
C     SUBROUTINE PAGEHD. COMMENT CARDS MAY HAVE ANY KEYPUNCH SYMBOL      0  801
C     IN CARD COLUMNS 1-78.                                              0  802
C     IF IT IS DESIRED TO HAVE ANY GIVEN COMMENT CARD PRINT ON A NEW     0  803
C     PAGE, SUPPLY THE LETTER P IN COLUMN 80 ON THAT CARD.               0  804
C     ROUTINE IS ENDED BY SUPPLYING A CARD WITH ZEROS IN COLUMNS 1 THRU 10.  0  805
C     CALLS FORMA SUBROUTINE PAGEHD.                                     0  806
C     CODED BY  RF HRUDA.  MARCH 1966.                                   0  807
C     LAST MODIFICATION BY  J ERNST.  JUNE 1971.                         0  808
C                                                                        0  809
 1001 FORMAT  (13A6,1X,A1)                                               0  810
 2001 FORMAT (///)                                                       0  811
 2002 FORMAT  (22X,13A6)                                                 0  812
C                                                                        0  813
      N = 0                                                              0  814
    1 READ (NIT,1001) (IREMRK(I),I=1,13),IPGHD                           0  815
      IF (IREMRK(1) .EQ. QUIT) RETURN                                    0  816
      N = N+1                                                            0  817
      IF (N.NE.1 .AND. IPGHD.NE.P) GO TO 2                               0  818
      CALL PAGEHD                                                        0  819
      WRITE (NOT,2001)                                                   0  820
      N = 1                                                              0  821
    2 IF (N .EQ. 50)  N = 0                                              0  822
      WRITE (NOT,2002) (IREMRK(I),I=1,13)                                0  823
      GO TO 1                                                            0  824
      END                                                                0  825
```

```
      SUBROUTINE PAGEHD                                                  0  826
C                                                     DEBUG #    8       0  827
      REAL*8 IRUNNO,DATE,UNAME,TITLE1,TITLE2,TIM                         0  828
      COMMON /ESTART/IRUNNO,DATE,NPAGE                                   0  829
      COMMON /ESTRT1/ UNAME(3),TITLE1(12),TITLE2(12)                     0  830
      DATA NOT / 6/                                                      0  831
C                                                                        0  832
C     BRINGS UP NEW PAGE AND PUTS HEADING AT TOP.                        0  833
C     BY COMBINED EFFORT OF BUDLEY, HRUDA, WOHLEN.  DEC 1969.            0  834
C     LAST REVISION BY JW ERNST.  SEP 1972.                              0  835
C                                                                        0  836
C         INTERNAL VARIABLES.  (TRANSFERRED THRU COMMON).                0  837
C     IRUNNO = RUN NUMBER. (A6 FORMAT)                                   0  838
C     DATE    = DATE. (A8 FORMAT)                                        0  839
C     NPAGE   = PAGE NUMBER.                                             0  840
C     UNAME   = USERS NAME. (3A6 FORMAT)                                 0  841
C     TITLE1 = FIRST  TITLE. (12A6 FORMAT)                               0  842
C     TITLE2 = SECOND TITLE. (12A6 FORMAT)                               0  843
C                                                                        0  844
 2001 FORMAT (9H1RUN NO. A6, 42X 5HDATE A8, 42X 8HPAGE NO. I4,           0  845
     *          / 55X 7HRUN BY 3A6, // 10X 12A6,10X15HCURRENT TIME = ,A8,  0  846
     *          / 10X 12A6,10X16HTHE CPU TIMER = ,1PE10.4)               0  847
C                                                                        0  848
      NPAGE = NPAGE + 1                                                  0  849
      CALL TIME (TIM)                                                    0  850
      CALL SECOND (SEC)                                                  0  851
      WRITE (NOT,2001) IRUNNO,DATE,NPAGE,UNAME,TITLE1,TIM,TITLE2,SEC     0  852
      RETURN                                                             0  853
      END                                                                0  854
      SUBROUTINE INTAPE (NTAPE,TAPEID)
C                                                        DEBUG #    9
      REAL*8 TAPEID,BUF,EOT
      DATA IZ1,BUF,EOT/1,0.D 0,3HEOT/
      DATA NOT / 6/
C
C  INITIALIZE TAPE FOR SUBROUTINE WTAPE.
C  CALLS FOR4A SUBROUTINE PAGEHD.
C  CODED BY  RF HRUDA.  JULY 1968.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NTAPE  = NUMBER OF TAPE. (E.G. 13).
C  TAPEID = TAPE IDENTIFICATION. (E.G. T1234). (A6 FORMAT).
C
 2001 FORMAT (//// 14H LOGICAL UNIT I2, 7H, TAPE A6,
     *                 23H, HAS BEEN INITIALIZED.)
C
      REWIND NTAPE
      WRITE (NTAPE) TAPEID,IZ1,EOT,(BUF,I=1,16)
      REWIND NTAPE
      CALL PAGEHD
      WRITE (NOT,2001) NTAPE,TAPEID
C
      RETURN
      END
```

```
      SUBROUTINE READ   (A,NR,NC,KR,KC)
C                                                        DEBUG #  10
      REAL*8 A,X,IREMRK,ANAME,IZZ,REWIND,LIST,TID,IDTCK,EOT
      DIMENSION A(KR,1),X(4),IREMRK(9)
      DATA NIT,NOT/5,6/
      DATA REWIND,LIST,EOT,IDOLAR,IASTRS/ 6HREWIND,4HLIST,3HEOT,1H$,1H*/
C
C  READ MATRIX OF REAL NUMBERS FROM CARDS OR TAPE AND PRINT IT. WRITE
C  MATRIX ON TAPE IF SO INDICATED (BY HAVING THE WRITE-TAPE NUMBER IN
C  COLUMNS 73-80).
C  THE EXPLANATION OF FORMATS USED BELOW IS ...
C     A - DENOTES ANY KEY PUNCH SYMBOL. (EG, A1/*C).
C     I - DENOTES AN INTEGER NUMBER. (EG, 436).
C     E - DENOTES A REAL NUMBER. (EG, 24.963).
C  **** CARD INPUT ****
C  FIRST CARD    - MATRIX NAME, NUMBER OF ROWS, NUMBER OF COLUMNS
C                  WITH A6,I4,I5 FORMAT.
C                - REMARKS IN COLUMNS 16-69. A-TYPE FORMAT.
C                - $ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.
C                - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR
C                  THE WORDS REWIND OR LIST, OR (WHEN $ IN COLUMN 72)
C                  THE WRITE-TAPE-ID (EG, T1234).
C                - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).
C  MIDDLE CARDS - DATA WITH FORMAT (2I5, 4E17).
C                - 1-ST I5 IS THE ROW NUMBER.
C                - 2-ND I5 IS THE COL NUMBER OF THE NEXT E17 FIELD.
C                - NEXT 4E17 ARE ELEMENTS OF THE MATRIX.
C  LAST CARD     - TEN ZEROS IN COLUMNS 1-10.
```

```
C    **** TAPE INPUT ****
C   ONE CARD   - MATRIX NAME, ZERO OR MINUS THE LOCATION NUMBER OF MATRIX
C                ON READ-TAPE, READ-TAPE NUMBER (IF MINUS, NO PRINTOUT),
C                MATRIX RUN NUMBER WITH A6,I4,I5,A6 FORMAT.
C              - READ-TAPE CONTROL IN COLUMNS 22-27. MAY BE BLANK, OR THE
C                WORDS REWIND OR LIST.
C              - REMARKS IN COLUMNS 28-69. A-TYPE FORMAT.
C              - $ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.
C              - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR
C                THE WORDS REWIND OR LIST, OR (WHEN $ IN COLUMN 72)
C                THE WRITE-TAPE-ID (EG. T1234).
C              - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG. 21).
C   CALLS FORMA SUBROUTINES INTAPE,LTAPE,PAGEHD,RTAPE,WRITE,WTAPE,ZZBOMB.
C   CODED BY  RF HRUDA.  JULY 1966.
C
C       SUBROUTINE ARGUMENTS
C   A      = OUTPUT  MATRIX READ FROM CARDS OR TAPE.
C   NR     = OUTPUT  NUMBER OF ROWS IN MATRIX A.
C   NC     = OUTPUT  NUMBER OF COLS IN MATRIX A.
C   KR     = INPUT   ROW DIMENSION OF A IN CALLING PROGRAM.
C   KC     = INPUT   COL DIMENSION OF A IN CALLING PROGRAM.
C
 1001 FORMAT (A6,I4,I5,9A6, 2XA1,A6,I2)
 1002 FORMAT (2I5,4D17.0)
 2001 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )
     *           2X 9A6,2X A1,A6,I4//)
 2002 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )
     *           3X 9HCONTINUED //)
 2003 FORMAT (// 1XA6,I4,I5,3X 9A6,2X A1,A6,I4)
 2004 FORMAT (1X 2I5,4D17.8)
 2005 FORMAT (13H0END OF READ.)
 2006 FORMAT (25H0SIZE OF MATRIX READ IS (I4,2H X I4,2H ) )
C
C   READ IN HEADER CARD.
      READ (NIT,1001) ANAME,N1,N2,IREMRK,I21,I22,NWTAPE
      IPRIN = 0
      IF (I21.EQ.IASTRS .OR. NWTAPE.GT.0 .OR. N1.LE.0) IPRIN = 1
      IF (IPRIN .EQ. 1) CALL PAGEHD
C
      IF (N1.LE.0)  GO TO 200
C
C   CARD READING SECTION.
      NR = N1
      NC = N2
      IF(IPRIN.EQ.1) WRITE (NOT,2001) ANAME,NR,NC,IREMRK,I21,I22,NWTAPE
      NERROR = 1
      IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999
      NLINE = 0
      DO 105 I=1,NR
      DO 105 J=1,NC
  105 A(I,J) = 0.D 0
  110 READ (NIT,1002) I,JJ,X
      IF (I.EQ.0 .AND. JJ.EQ.0)  GO TO 300
      NERROR = 2
      IF (I.LE.0 .OR. I.GT.NR .OR. JJ.LE.0 .OR. JJ.GT.NC) GO TO 998
      JE = JJ+3
      IF (JE.LE.NC) GO TO 115
      JX = NC-JJ+2
      NERROR = 3
      DO 112 J=JX,4
  112 IF (X(J) .NE. 0.D 0) GO TO 998
      JE = NC
  115 N = 0
      DO 120 J=JS,JE
      N = N+1
  120 A(I,J) = X(N)
      NLINE = NLINE+1
      IF (NLINE.LE.47)  GO TO 125
      IF (IPRIN .EQ. 1) CALL PAGEHD
      IF (IPRIN .EQ. 1) WRITE (NOT,2002) ANAME,NR,NC
      NLINE = 1
  125 IF (IPRIN .EQ. 1) WRITE (NOT,2004) I,JS,(A(I,J),J=JS,JE)
      GO TO 110
C
C  TAPE READING SECTION.
  200 WRITE (NOT,2003) ANAME,N1,N2,IREMRK,I21,I22,NWTAPE
      NRTAPE = IABS(N2)
      IF (IREMRK(2) .EQ.   REWIND) REWIND NRTAPE
      IF (IREMRK(2) .EQ.   LIST)   CALL LTAPE (NRTAPE)
      IF (N1.EQ.0)  GO TO 250
C  POSITION NRTAPE.
      READ  (NRTAPE) TID,LN,IEOTCK
      NUM = LN+N1
      IF (NUM) 205,220,225
  205 NERROR = 4
      IF (IEOTCK .EQ.   EOT)  GO TO 997
      READ  (NRTAPE)
      NUM = -NUM-1
      IF (NUM.EQ.0)  GO TO 250
      DO 210 L=1,NUM
      READ  (NRTAPE) TID,LN,IEOTCK
      NERROR = 5
      IF (IEOTCK .EQ.   EOT)  GO TO 997
  210 READ  (NRTAPE)
      GO TO 250
  220 BACKSPACE NRTAPE
      GO TO 250
  225 REWIND NRTAPE
      NUM = (-N1-1)*2
      IF (NUM.EQ.0)  GO TO 250
      DO 230 L=1,NUM
  230 READ  (NRTAPE)
  250 CALL RTAPE  (IREMRK(1),ANAME,A,NR,NC,KR,KC,NRTAPE)
      WRITE (NOT,2006) NR,NC
      IF (N2 .GT. 0)  CALL WRITE  (A,NR,NC,ANAME,KR)
C
C  TAPE WRITING SECTION.
  300 IF (NWTAPE.LE.0)  GO TO 400
      IF (I21 .EQ. IDOLAR)  CALL INTAPE (NWTAPE,I22)
      IF (I22 .EQ.   REWIND) REWIND NWTAPE
      CALL WTAPE  (A,NR,NC,ANAME,KR,NWTAPE)
      IF (I22 .EQ.   LIST)   CALL LTAPE (NWTAPE)
C
  400 IF (IPRIN .EQ. 1) WRITE (NOT,2005)
      RETURN
C
  997 CALL LTAPE  (NRTAPE)
      GO TO 999
  998 WRITE (NOT,2004) I,JS,X
  999 WRITE (NOT,2010) NERROR
```

```
 2010 FORMAT (1H1,42HPROGRAM STOPPED, ERROR IN SUBROUTINE READ,
     *       13H NERROR = ,I3)
      STOP
      END




      SUBROUTINE RTAPE (IARUNO,IANAME, A,NRA,NCA, KR,KC,NTAPE)
C
C                                                     DEBUG #  11
      REAL*8 A,IARUNO,IANAME,TAPEID,IEOTCK,ITRUNO,ITNAME,
     *        DATE,ITYPE,DENSE,EOT
      DIMENSION A(KR,1)
      DATA NOT / 6 /
      DATA DENSE,EOT / 5HDENSE,3HEOT /
C
C   READ MATRIX A FROM TAPE BY IDENTIFICATION OF IARUNO,IANAME.
C   CALLS FORMA SUBROUTINES ETAPE,PAGEHD,ZZBOMB.
C   CODED BY WA BENFIELD.  JUNE 1966.
C   LAST REVISION BY R F BRUDA.  SEPTEMBER 1971.
C
C       SUBROUTINE ARGUMENTS
C   IARUNO = INPUT  RUN NUMBER OF MATRIX A. (A6 FORMAT).
C   IANAME = INPUT  MATRIX IDENTIFICATION.  (A6 FORMAT).
C   A      = OUTPUT MATRIX READ FROM TAPE. SIZE(NRA,NCA).
C   NRA    = OUTPUT NUMBER OF ROWS OF MATRIX A. WILL BE READ FROM TAPE.
C   NCA    = OUTPUT NUMBER OF COLS OF MATRIX A. WILL BE READ FROM TAPE.
C   KR     = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C   KC     = INPUT  COL DIMENSION OF A IN CALLING PROGRAM.
C   NTAPE  = INPUT  NUMBER OF TAPE. (E.G. 10).
C
 3001 FORMAT (29H1RTAPE CANNOT FIND   RUNNO = A6 /
     *          21X 8HANAME = A6 / 29X 6H------ )
C
      NTIME = 0
C   SEARCH TAPE FOR CORRECT HEADING.
    5 READ (NTAPE) TAPEID,LN,IEOTCK,ITRUNO,ITNAME,NRA,NCA,DATE,ITYPE,NNZ
      IF (ITRUNO .EQ. IARUNO .AND. ITNAME .EQ. IANAME) GO TO 10
      IF (IEOTCK .EQ.   EOT) GO TO 20
      READ  (NTAPE)
      GO TO 5
C
C   MATRIX HAS BEEN FOUND.
   10                                                 NERROR=1
      IF (ITYPE .NE.   DENSE   .AND. NNZ .NE. 0) GO TO 999
                                                      NERROR=2
      IF (NRA.GT.KR .OR. NCA.GT.KC)  GO TO 999
      READ  (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
      RETURN
C
C   MATRIX CANNOT BE FOUND. SEARCH TAPE ONCE MORE.
   20 NTIME = NTIME+1
                                                      NERROR=3
      IF (NTIME .EQ. 2) GO TO 998
      REWIND NTAPE
      GO TO 5
C
```

```
  998 WRITE (NCT,3001) IARUNO,IANAME
  999 CALL ETAPE  (NTAPE)
      WRITE (NCT,3002) NERROR
 3002 FORMAT (1H1,43HERROR IN SUBROUTINE RTAPE, PROGRAM STOPPED,
     *        10H NERROR = ,I3)
      STOP
      END
```

```
      SUBROUTINE WRITE  (A,NR,NC,ANAME,KR)
C                                                     DEBUG #  12
      REAL*8 A,ANAME
      DIMENSION A(KR,1)
      DATA NCT / 6/
C
C  WRITE MATRIX OF REAL NUMBERS ON PAPER.
C  REQUIRES 123 COLUMN (MINIMUM) PRINTER.
C  UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RL WOHLEN.  DECEMBER 1968.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  A      = MATRIX TO BE PRINTED. SIZE(NR,NC).
C  NR     = NUMBER OF ROWS IN MATRIX A.
C  NC     = NUMBER OF COLS IN MATRIX A.
C  ANAME  = MATRIX IDENTIFICATION. (A6 FORMAT).
C  KR     = ROW DIMENSION OF A IN CALLING PROGRAM.
C
 2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //
     *        10X,10(7X,1H( I2,1H))/)
 2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H )
     *         3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/)
 2030 FORMAT (1X,2I5,2X,1P10D11.3)
 2040 FORMAT (14H0END OF WRITE.)
C
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.
      CALL PAGEHD
      WRITE (NCT,2010) ANAME,NR,NC,(L,L=1,10)
      NLINE = 0
C
      DO 60 I=1,NR
      NZERO = 0
      JS = 1
   10 JE = JS+9
      IF (JE .GT. NC) JE=NC
C  SEE IF ELEMENTS ARE ZERO.
      DO 20 J=JS,JE
   20 IF (A(I,J) .NE. 0.D 0) GO TO 30
      GO TO 40
   30 NLINE = NLINE+1
      IF (NLINE .LE. 44) GO TO 35
      CALL PAGEHD
      WRITE (NCT,2020) ANAME,NR,NC,(L,L=1,10)
      NLINE = 1
   35 WRITE (NCT,2030) I,JS,(A(I,J), J=JS,JE)
      NZERO = 1
   40 IF (JE .EQ. NC) GO TO 50
      JS = JS+10
      GO TO 10
C   SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS
C   AND SOMETHING HAS BEEN WRITTEN.
   50 IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR)  GO TO 60
      NLINE = NLINE+1
      WRITE (NOT,2030)
   60 CONTINUE
C
      WRITE (NOT,2040)
      RETURN
      END
```

```
      SUBROUTINE WTAPE  (A,NRA,NCA,ANAME,KR,NTAPE)
C                                                       DEBUG #  13
      REAL*8 A,ANAME,IRUNNO,DATE,UNAME,TITLE1,TITLE2,
     *       EOF,EOT,DENSE,TAPEID,IEOTCK
      DIMENSION  A(KR,1)
      COMMON /ESTART/IRUNNO,DATE,NPAGE
      COMMON /ESTRT1/ UNAME(3),TITLE1(12),TITLE2(12)
      DATA EOF,EOT,DENSE/0.D0,3HEOT,5HDENSE/
      DATA NOT / 6/
C
C   WRITE MATRIX A ON TAPE.
C   INITIALIZE TAPE WITH SUBROUTINE INTAPE.
C   REWIND TAPE BEFORE FIRST USE OF THIS SUBROUTINE.
C   NOTE...THIS ROUTINE IS DESIGNED SPECIFICALLY FOR WRITING ON A DISK
C
C          (EG CDC-6400 DISK). USING THIS ROUTINE TO WRITE ON A PHYSICAL
C          TAPE DIRECTLY (IE WITHOUT USING THE DISK AS AN INTERMEDIARY)
C          WILL PROBABLY GIVE POOR RESULTS (DUE TO THE TOLERANCE
C          CHARACTERISTICS OF A TAPE DRIVE) AND SHOULD BE AVOIDED IF AT
C          ALL POSSIBLE.
C       ...THE CDC-6400 DISK IS AUTOMATICALLY ENDFILED AFTER EACH WRITE.
C   CODED BY    W A BENFIELD.  MARCH 1968.
C   REVISED BY R A PHILIPPUS.  APRIL 1969.
C   REVISED BY  RF PRODA.  NOVEMBER 1973.
C
C       SUBROUTINE ARGUMENTS (ALL INPUT)
C   A       = MATRIX TO BE WRITTEN ON TAPE. SIZE(NRA,NCA).
C   NRA     = NUMBER OF ROWS OF MATRIX A.
C   NCA     = NUMBER OF COLS OF MATRIX A.
C   ANAME   = MATRIX IDENTIFICATION. (A6 FORMAT).
C   KR      = ROW DIMENSION OF A IN CALLING PROGRAM.
C   NTAPE   = NUMBER OF TAPE. (E.G. 10).
C
C       INTERNAL VARIABLES THAT ARE PUT ON TAPE (TRANSFERRED THRU COMMON).
C   IRUNNO IS RUN NUMBER OF PROBLEM. (A6 FORMAT).
C   DATE IS DATE. (A6 FORMAT).  FOR EXAMPLE  13FE69.
C
      IF (NRA .LT. 1 .OR. NCA .LT. 1) GO TO 999
C
C   SEARCH TAPE FOR END OF WRITTEN DATA.
   10 READ (NTAPE) TAPEID,C,IEOTCK
```

```
      IF (IEOTCK .EQ.   EOT) GO TO 20
      READ (NTAPE)
      GO TO 10
C
C   END OF WRITTEN DATA HAS BEEN FOUND.
   20 BACKSPACE NTAPE
      WRITE (NTAPE) TAPEID,LN,BUF,IRUNNO,ANAME,NRA,NCA,DATE,DENSE,
     *              (BUF,I=1,10)
      WRITE (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
      LN = LN + 1
      WRITE (NTAPE) TAPEID,LN,EOT,(BUF,I=1,16)
      BACKSPACE NTAPE
      RETURN
C
  999 WRITE (NOT,1000)
 1000 FORMAT (1H1,43HERROR IN SUBROUTINE WTAPE, PROGRAM STOPPED.)
      STOP
      END
```

```
      SUBROUTINE LTAPE (NTAPE)
C                                                       DEBUG #  14
      REAL*8 TAPEID,IRUNNO,ANAME,IEOTCK,DATE,ITYPE,ICHK,EOT,
     *    DENSE,SPARSE,SPART
      DATA NOT / 6/
      DATA EOT,DENSE,SPARSE,SPART/ 3HEOT,5HDENSE,6HSPARSE,5HSPART/
C
C   LIST HEADINGS OF MATRICES ON TAPE.
C   CALLS FCH4A SUBROUTINE PAGEHD.
C   CODED BY  HF HRUBA.  JULY 1968.   REVISED NOVEMBER 1970.
C   REVISED BY R A PHILIPPUS.  APRIL 1969.
C
C       SUBROUTINE ARGUMENTS (ALL INPUT)
C   NTAPE  = NUMBER OF TAPE. (E.G. 10).
C
 2001 FORMAT (//30X33HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6)
 2002 FORMAT (//30X33HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6,
     *          12H (CONTINUED))
 2003 FORMAT (27X69(1H-)/27X3HNO.3X7HRUN NO.4X4HNAME3X5HNROWS4X5HNCOLS4X
     *          4HDATE6X3HNNZ3X9HPARTITION/
     *            27X3H---3X6H------4X6H------4X5H-----
     *            4X5H-----3X6H------5X3H---3X9H--------- / )
 2004 FORMAT (25XI5,3XA6,4XA6,3XI5,4XI5,4XA6,3XI3,3XI4,1H/I4)
 2005 FORMAT (//27X12HEND OF LIST.)
C
      REWIND NTAPE
      READ  (NTAPE) TAPEID
      REWIND NTAPE
      L=0
C
   12 CALL PAGEHD
      IF(L .EQ. 0)  WRITE (NOT,2001) NTAPE,TAPEID
      IF(L .NE. 0)  WRITE (NOT,2002) NTAPE,TAPEID
      WRITE (NOT,2003)
      NLINE=1
   13 L=L+1
```

```
      READ (NTAPE) TAPEID,LN,IEOTCK,IRUNNO,ANAME,NR,NC,DATE,ITYPE,NNZ,     0 1245
     *              NP,NPT                                                 0 1246
      IF (L .EQ. 1)  ICHK = IRUNNO                                         0 1247
      IF (ICHK .EQ. IRUNNO) GO TO 15                                       0 1248
      NLINE=NLINE+1                                                        0 1249
      WRITE (NOT,2004)                                                     0 1250
      ICHK = IRUNNO                                                        0 1251
   15 IF (IEOTCK .EQ. EOT) GO TO 30                                        0 1252
      READ  (NTAPE)                                                        0 1253
      IF (ITYPE .EQ.    DENSE ) WRITE (NOT,2004)                           0 1254
     *                          LN,IRUNNO,ANAME,NR,NC,DATE                 0 1255
      IF (ITYPE .EQ.    DENSE ) GO TO 20                                   0 1256
      IF (ITYPE .EQ.    SPARSE) WRITE (NOT,2004)                           0 1257
     *                          LN,IRUNNO,ANAME,NR,NC,DATE,NNZ             0 1258
      IF (ITYPE .EQ.    SPARSE) GO TO 20                                   0 1259
      IF (ITYPE .EQ.    SPART ) WRITE (NOT,2004)                           0 1260
     *                          LN,IRUNNO,ANAME,NR,NC,DATE,NNZ,NP,NPT      0 1261
      IF (ITYPE .EQ.    SPART ) GO TO 20                                   0 1262
      WRITE (NOT,2004) LN,IRUNNO,ANAME,NR,NC,ITYPE                         0 1263
   20 NLINE=NLINE+1                                                        0 1264
      IF(NLINE.GT.43) GO TO 12                                             0 1265
      GO TO 15                                                             0 1266
C                                                                          0 1267
   30 WRITE (NOT,2004) LN,IEOTCK                                           0 1268
      WRITE (NOT,2005)                                                     0 1269
      REWIND NTAPE                                                         0 1270
      RETURN                                                               0 1271
      END                                                                  0 1272
```

```
      SUBROUTINE MSYDOC (NBUG)                                             0 1273
C                                                      DEBUG #   15        0 1274
      IMPLICIT REAL*8(A-H,O-Z)                                             0 1275
C                                                                          0 1276
            COMMON /MAXMUM/                                                0 1277
     *      NJMAX,NMMAX,NSPMAX,NMWMAX,NMWBOD,NMDJJJ,KMJ,KY,KU              0 1278
            COMMON /NHNS  /                                                0 1279
     *      NHPCI( 6), NSPCI( 6)                                          12 1280
            COMMON /NUMBRS/                                                0 1281
     *      ZRO,ONE,TWO,TRES                                               0 1282
            COMMON /SPECIF/                                                0 1283
     *      BETAH(6, 6),BETAHO(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),     16 1284
     *      BH(3,30),BS(3,30),IMO(3, 5),NMOW(6, 6),IFTSMW(15),            17 1285
     *      NJ,NM,NSPT,NOFMO,NDELTA,ITOPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 1286
     *      LOCO(14),LENO(14),NO,NBETA,NLAM,NEQ                           19 1287
            COMMON /SUMMRY/                                                0 1288
     *      ASUMRY(15,6),ISUMRY(15,3),KSUMRY                              98 1289
            COMMON /TAPENO/                                                0 1290
     *      NTAPE1,NTAPE2,NTAPE3                                           0 1291
C                                                                          0 1292
      DIMENSION A( 42, 42),B( 42, 42),IV( 42),JV( 42),C(6,6),PHR6(6,6),   61 1293
     *   BC(9,12),WS1(12,12),WS2(12,12),CM2( 42),OMSA2(18),JDOF( 7,6)     62 1294
C                                                                          0 1295
      DATA NIT,NOT,KAB,KJDOF/ 5,6, 42, 7 /                                63 1296
 1001 FORMAT (16I5)                                                        0 1297
 1002 FORMAT (8D10.3)                                                      0 1298
```

```
3001 FORMAT (//15X30HSUMMARY OF INPUT DATA FOR BODY,I3,
     * 44H WHICH IS FLEXIBLE W/CONSISTENT MASS MATRIX.//
     * 3X49HTHE INTEGER PARAMETERS--- IFRBM,IFDIAK,IFDIAD ARE5XI2,1H,
     *    5XI2,1H,5XI2,//
     *  3X25HTHE JDOF TABLE FOLLOWS---)
3002 FORMAT (//3X30HTHE MODE SELECTION VECTOR FOLLOWS---)
3003 FORMAT (//3X12HFOR BODY NO.,I3,25H THE POSITION VECTOR FROM
     *25H THE BODY ORIGIN TO JOINT,I4,3H IS. /
     *  10X 4HX = 1PD10.3,5X5H Y = 1PD10.3,5X5H Z = 1PD10.3)
3004 FORMAT (//5X46HTHE CONSISTENT, REPARTITIONED MASS MATRIX IS--)
3005 FORMAT (//5X36HTHE REPARTITIONED MODAL MATRIX IS---)
3006 FORMAT (//5X44HTHE  -UNDEFORMED-  INERTIA MATRIX (MO) IS---)
3007 FORMAT (//5X27HTHE  A  COEFFICIENTS ARE---)
3008 FORMAT (//5X27HTHE  B  COEFFICIENTS ARE---)
3009 FORMAT (//5X31HTHE  COFXY  COEFFICIENTS ARE---)
3010 FORMAT (//5X31HTHE  COFXZ  COEFFICIENTS ARE---)
3011 FORMAT (//5X31HTHE  COFYZ  COEFFICIENTS ARE---)
3012 FORMAT (//5X31HTHE  C11    COEFFICIENTS ARE---)
3013 FORMAT (//5X31HTHE  C22    COEFFICIENTS ARE---)
3014 FORMAT (//5X31HTHE  C33    COEFFICIENTS ARE---)
3015 FORMAT (//5X31HTHE  C12    COEFFICIENTS ARE---)
3016 FORMAT (//5X31HTHE  C13    COEFFICIENTS ARE---)
3017 FORMAT (//5X31HTHE  C23    COEFFICIENTS ARE---)
3018 FORMAT (//5X33HTHE  MODAL STIFFNESS MATRIX IS---)
3019 FORMAT (//5X31HTHE  MODAL DAMPING MATRIX IS---)
3020 FORMAT (//5X50HTHE  INITIAL MODAL COORDINATE DISPLACEMENTS ARE---)
3021 FORMAT (//5X47HTHE  INITIAL MODAL COORDINATE VELOCITIES ARE---)
3022 FORMAT (//5X 9HFOR BODY I3,29H THE P-Q HINGE NO., THE EULER
     * 57H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE P-Q./
     *  5X54H HINGE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS
     *     49HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/
     * 5X44H POSITION THE HINGE TRIAD WRT THE BODY TRIAD)
3023 FORMAT (//5X 9HFOR BODY I3,32H THE SENSOR POINT NO., THE EULER
     *60H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE SENSOR./
     *5X53HPOINT APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS
     *     49HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/
     * 5X45H POSITION THE SENSOR TRIAD WRT THE BODY TRIAD)
C
      REWIND NTAPE2
      KMO = NMOBOD
      CALL ZERO   (BC,9,KMO,9)
      READ (NIT,1001) IFRBM,IDIAK,IDIAD
      CALL READIM (JDOF,NX,ND,KJDOF,6)
      IC = 0
      DO 2 J=1,6
      DO 2 I=1,NX
      NDF = JDOF(I,J)
      IC = IC + 1
    2 IV(NDF) = IC
      CALL READIM (JV,N1,N2,1,KAB)
      NY = NX
      NZ = NX
      NMO = 0
      CALL PAGEHD
      WRITE (NOT,3001) NBOD,IFRBM,IDIAK,IDIAD
      CALL WRITIS (JDOF,NX,6,KJDOF)
      WRITE (NOT,3002)
      CALL WRITIS (JV,1,N2,1)
C
      DO 3 I=1,N2
      IF (JV(I) .EQ. 0) GO TO 3
      NMO = NMO + 1
    3 CONTINUE
      IF (NMO .GT. KMO + 6) GO TO 999
      CALL READ   (A,NRA,NCA,KAB,KAB)
      CALL REVISE (A,IV,IV,B,NRA,NCA,NRA,NRA,KAB,KAB)
      WRITE (NTAPE2) ((B(I,J),I=1,NRA),J=1,NRA)
      REWIND NTAPE2
C
      CALL READ   (A,NRA,NMOT,KAB,KAB)
      IF (IDIAK .EQ. 0 .AND. IDIAD .EQ. 0) GO TO 11
      CALL READ   (OM2,N1,N2,1,KAB)
   11 NE = NMO - 6
      CALL REVISE (A,IV,JV,B,NRA,NMOT,NRA,NMO,KAB,KAB)
      IF (IDIAK .EQ. 0 .AND. IDIAD .EQ. 0) GO TO 12
      CALL REVISE (OM2,1,JV,OMGA2,1,N2,1,NMO,1,1)
   12 IF (IFREM .EQ. 0) GO TO 5
      READ (NIT,1001) JTYPCL
      READ (NIT,1002) (OM2(J),J=1,3)
      WRITE (NOT,3003) NLOD,JTYPCL,OM2(1),OM2(2),OM2(3)
      JRB = JTYPCL - NX
      DO 4 I=1,6
      JRB = JRB + NX
      DO 4 J=1,6
    4 PHR6(I,J) = B(JRB,J)
      CALL GAUSS1 (PHR6,C,6,6)
      CALL ZERO   (PHR6(4,4),3,3,6)
      CALL UNITY (PHR6(1,4),3,6)
      CALL UNITY (PHR6(4,1),3,6)
      CALL SKEWV3 (OM2,PHR6,1,6)
      CALL MULTA  (C,PHR6,6,6,6,6,6)
      CALL MULTA  (B,C,NRA,6,6,KAB,6)
C
    5 READ (NTAPE2) ((A(I,J),I=1,NRA),J=1,NRA)
      REWIND NTAPE2
      WRITE (NOT,3004)
      CALL WRITES (A   ,NRA,NRA,KAB)
      WRITE (NOT,3005)
      CALL WRITES (B   ,NRA,NMO,KAB)
C
      CALL BTABA (A,B,NRA,NMO,KAB,KAB)
      WRITE (NOT,3006)
      CALL WRITES (A   ,NMO,NMO,KAB)
      WRITE (NTAPE1) ((A(I,J),J=1,NMO),I=1,NMO)
      DO 25 J=1,NE
      JP6 = J + 6
   25 OM2(J) = A(JP6,JP6)
C
      READ (NTAPE2) ((A(I,J),I=1,NRA),J=1,NRA)
      REWIND NTAPE2
      NRP = 3*NX
      CALL MULTA (A,B,NRP,NRA,NMO,KAB,KAB)
      CALL ZERO   (BC,9,NE,9)
      DO 15 J=1,NE
      K = 6 + J
      DO 15 IX = 1,NX
      IY = IX + NX
      IZ = IY + NX
      BC(1,J) = BC(1,J) + A(IZ,4)*B(IY,K) - A(IY,4)*B(IZ,K)
      BC(2,J) = BC(2,J) + A(IZ,5)*B(IY,K) - A(IY,5)*B(IZ,K)
      BC(3,J) = BC(3,J) + A(IZ,6)*B(IY,K) - A(IY,6)*B(IZ,K)
      BC(4,J) = BC(4,J) + A(IX,4)*B(IZ,K) - A(IZ,4)*B(IX,K)
      BC(5,J) = BC(5,J) + A(IX,5)*B(IZ,K) - A(IZ,5)*B(IX,K)
      BC(6,J) = BC(6,J) + A(IX,6)*B(IZ,K) - A(IZ,6)*B(IX,K)
      BC(7,J) = BC(7,J) + A(IY,4)*B(IX,K) - A(IX,4)*B(IY,K)
      BC(8,J) = BC(8,J) + A(IY,5)*B(IX,K) - A(IX,5)*B(IY,K)
      BC(9,J) = BC(9,J) + A(IY,6)*B(IX,K) - A(IX,6)*B(IY,K)
   15 CONTINUE
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,9)
      WRITE (NCT,3007)
      CALL WRITES (BC  ,  9,NE  ,  9)
C
      CALL ZERO  (BC,9,NE,9)
      DO 16 J=1,NE
      K = 6 + J
      DO 16 IX=1,NX
      IY = IX + NX
      IZ = IY + NX
      BC(1,J) = BC(1,J) + A(IZ,1)*B(IY,K) - A(IY,1)*B(IZ,K)
      BC(2,J) = BC(2,J) + A(IX,2)*B(IZ,K) - A(IZ,2)*B(IX,K)
      BC(3,J) = BC(3,J) + A(IY,3)*B(IX,K) - A(IX,3)*B(IY,K)
      BC(4,J) = BC(4,J) + A(IZ,1)*B(IX,K) - A(IX,1)*B(IZ,K)
     *                  + A(IY,2)*B(IZ,K) - A(IZ,2)*B(IY,K)
      BC(5,J) = BC(5,J) + A(IX,1)*B(IY,K) - A(IY,1)*B(IX,K)
     *                  + A(IY,3)*B(IZ,K) - A(IZ,3)*B(IY,K)
      BC(6,J) = BC(6,J) + A(IX,2)*B(IY,K) - A(IY,2)*B(IX,K)
     *                  + A(IZ,3)*B(IX,K) - A(IX,3)*B(IZ,K)
   16 CONTINUE
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,6)
      WRITE (NCT,3008)
      CALL WRITES (BC  ,  6,NE  ,  9)
C
      CALL ZERO  (WS1,NE,NE,KMD)
      DO 17 I=1,NE
      KI = 6 + I
      DO 17 J=1,NE
      KJ = 6 + J
      DO 17 IX=1,NX
      IY = IX + NX
      IZ = IY + NX
      WS1(I,J) = WS1(I,J) + B(IX,KI)*A(IY,KJ) - B(IY,KI)*A(IX,KJ)
   17 CONTINUE
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3009)
      CALL WRITES (WS1,NE  ,NE  ,KMD)
C
      CALL ZERO  (WS1,NE,NE,KMD)
      DO 18 I=1,NE
      KI = 6 + I
      DO 18 J=1,NE
      KJ = 6 + J
      DO 18 IX=1,NX
      IY = IX + NX
      IZ = IY + NX
      WS1(I,J) = WS1(I,J) + B(IZ,KI)*A(IX,KJ) - B(IX,KI)*A(IZ,KJ)
   18 CONTINUE
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3010)
      CALL WRITES (WS1,NE  ,NE  ,KMD)
C
```

```
      CALL ZERO   (WS1,NE,NE,KMO)
      DO 15 I=1,NE
      KI = 6 + I
      DO 15 J=1,NE
      KJ = 6 + J
      DO 15 IX=1,NX
      IY = IX + NX
      IZ = IY + NX
      WS1(I,J) = WS1(I,J) + B(IY,KI)*A(IZ,KJ) - B(IZ,KI)*A(IY,KJ)
   15 CONTINUE
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3011)
      CALL WRITES (WS1,NE ,NE ,KMO)
C
      LX = 1
      LY = LX + NX
      LZ = LY + NY
      S1 = ONE
C
      READ (NTAPE2) ((A(I,J),I=1,NRA),J=1,NRA)
C
C
      CALL ZERO   (WS1,NE,NE,KMO)
      CALL PR3 (A(LZ,LZ),B(LY,7),B(LY,7),WS2,WS1, S1,NZ,NZ,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LZ,LY),B(LZ,7),B(LY,7),WS2,WS1,-S1,NZ,NY,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LY,LZ),B(LY,7),B(LZ,7),WS2,WS1,-S1,NY,NZ,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LY,LY),B(LZ,7),B(LZ,7),WS2,WS1, S1,NY,NY,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3012)
      CALL WRITES (WS1,NE ,NE ,KMO)
      CALL ZERO   (WS1,NE,NE,KMO)
      CALL PR3 (A(LX,LX),B(LZ,7),B(LZ,7),WS2,WS1, S1,NX,NX,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LX,LZ),B(LX,7),B(LZ,7),WS2,WS1,-S1,NX,NZ,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LZ,LX),B(LZ,7),B(LX,7),WS2,WS1,-S1,NZ,NX,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LZ,LZ),B(LX,7),B(LX,7),WS2,WS1, S1,NZ,NZ,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3013)
      CALL WRITES (WS1,NE ,NE ,KMO)
      CALL ZERO   (WS1,NE,NE,KMO)
      CALL PR3 (A(LY,LY),B(LX,7),B(LX,7),WS2,WS1, S1,NY,NY,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LY,LX),B(LY,7),B(LX,7),WS2,WS1,-S1,NY,NX,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LX,LY),B(LX,7),B(LY,7),WS2,WS1,-S1,NX,NY,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      CALL PR3 (A(LX,LX),B(LY,7),B(LY,7),WS2,WS1, S1,NX,NX,NE,NE,
     *  KAE,KAE,KAB,KJDUF,KMO)
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3014)
      CALL WRITES (WS1,NE ,NE ,KMO)
      CALL ZERO   (WS1,NE,NE,KMO)
      CALL PR3 (A(LZ,LX),B(LZ,7),B(LY,7),WS2,WS1,-S1,NZ,NX,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      CALL PR3 (A(LZ,LZ),B(LX,7),B(LY,7),WS2,WS1, S1,NZ,NZ,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      CALL PR3 (A(LY,LX),B(LZ,7),B(LZ,7),WS2,WS1, S1,NY,NX,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      CALL PR3 (A(LY,LZ),B(LX,7),B(LZ,7),WS2,WS1,-S1,NY,NZ,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3015)
      CALL WRITES (WS1,NE  ,NE  ,KMO)
      CALL ZERO   (WS1,NE,NE,KMO)
      CALL PR3 (A(LZ,LY),B(LX,7),B(LY,7),WS2,WS1,-S1,NZ,NY,NE,NE,
     *   KAE,KAB,KAB,KJDOF,KMO)
      CALL PR3 (A(LZ,LX),B(LY,7),B(LY,7),WS2,WS1, S1,NZ,NX,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      CALL PR3 (A(LY,LY),B(LX,7),B(LZ,7),WS2,WS1, S1,NY,NY,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      CALL PR3 (A(LY,LX),B(LY,7),B(LZ,7),WS2,WS1,-S1,NY,NX,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3016)
      CALL WRITES (WS1,NE  ,NE  ,KMO)
      CALL ZERO   (WS1,NE,NE,KMO)
      CALL PR3 (A(LX,LY),B(LX,7),B(LZ,7),WS2,WS1,-S1,NX,NY,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      CALL PR3 (A(LX,LX),B(LY,7),B(LZ,7),WS2,WS1, S1,NX,NX,NE,NE,
     *   KAE,KAB,KAB,KJDOF,KMO)
      CALL PR3 (A(LZ,LY),B(LX,7),B(LX,7),WS2,WS1, S1,NZ,NY,NE,NE,
     *   KAE,KAB,KAB,KJDOF,KMO)
      CALL PR3 (A(LZ,LX),B(LY,7),B(LX,7),WS2,WS1,-S1,NZ,NX,NE,NE,
     *   KAE,KAE,KAB,KJDOF,KMO)
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3017)
      CALL WRITES (WS1,NE  ,NE  ,KMO)
C
      IF (IDIAK .EQ. 1) GO TO 50
      CALL READ (A,NRA,NCA,KAB,KAB)
      CALL BTABA (A,B(1,7),NRA,NE,KAB,KAB)
      GO TO 51
   50 CALL ZERO (A,NE,NE,KAB)
      DO 55 J=1,NE
      JP6 = J + 6
   55 A(J,J) = OM2(J)*OMGA2(JP6)
   51 WRITE (NTAPE1) ((A(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3018)
      CALL WRITES (A   ,NE  ,NE  ,KAB)
      IF (IDIAD .EQ. 1) GO TO 60
      CALL READ (A,NRA,NCA,KAB,KAB)
      CALL BTABA (A,B(1,7),NRA,NE,KAB,KAB)
      GO TO 61
   60 CALL ZERO (A,NE,NE,KAB)
      DO 65 J=1,NE
      JP6 = J + 6
   65 OMGA2(JP6) = TWO*OM2(J)*DSQRT(OMGA2(JP6))
      READ (NIT,1002) (OM2(J),J=1,NE)
      DO 66 J=1,NE
      JP6 = J + 6
   66 A(J,J) = OM2(J)*OMGA2(JP6)
   61 WRITE (NTAPE1) ((A(I,J),J=1,NE),I=1,NE)
      WRITE (NCT,3019)
      CALL WRITES (A     ,NE ,NE ,KAB)
C
      READ (NIT,1002) (OM2(J),J=1,NE)
      WRITE (NTAPE1) (OM2(J),J=1,NE)
      WRITE (NCT,3020)
      CALL WRITES (OM2,  1,NE ,  1)
      READ (NIT,1002) (OM2(J),J=1,NE)
      WRITE (NTAPE1) (OM2(J),J=1,NE)
      WRITE (NCT,3021)
      CALL WRITES (OM2,  1,NE ,  1)
C
      NHB = NHPOI(NBOD)
      NSB = NSPOI(NBOD)
CCC   NHB IS NO. OF P-G HINGES ON THE BODY, NOT TO INCLUDE HINGE 1
      WRITE (NTAPE1) NHB
      DO 110 L=1,NHB

      READ (NIT,1001) NOH,ITYPE,JOINT
      ISUMRY(L,1) = NOH
      ISUMRY(L,2) = ITYPE
      ISUMRY(L,3) = JOINT
      IF (NOH .LT. 2 .OR. NOH .GT. NH) GO TO 998
      LR = 6*(NOH - 2) + 1
      LJ = 7*(NOH - 2) + 1
      IF (ITOPOL(1,NOH) .EQ. NBOD) GO TO 112
      IF (ITOPOL(2,NOH) .NE. NBOD) GO TO 998
      LR = LR + 1
      LJ = LJ + 1
  112 JHX = JOINT
      JHY = JHX + NX
      JHZ = JHY + NX
      JSX = JHZ + NX
      JSY = JSX + NX
      JSZ = JSY + NX
      GH(1,LJ) = B(JHY,3)
      GH(2,LJ) = B(JHZ,1)
      GH(3,LJ) = B(JHX,2)
      READ (NIT,1002) (OM2(J),J=1,3)
      ASUMRY(L,1) = OM2(1)
      ASUMRY(L,2) = OM2(2)
      ASUMRY(L,3) = OM2(3)
      CALL ROTTR  (3,ITYPE,OM2,RH(1,1,LR),DUM,DUM)
      DO 115 J=1,NE
      JPG = J + 6
      BC(1,J) = B(JHX,JPG)
      BC(2,J) = B(JHY,JPG)
      BC(3,J) = B(JHZ,JPG)
      BC(4,J) = B(JSX,JPG)
      BC(5,J) = B(JSY,JPG)
  115 BC(6,J) = B(JSZ,JPG)
      WRITE (NTAPE1) NOH
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,3)
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=4,6)
  110 CONTINUE
      WRITE (NCT,3022) NBOD
      CALL WRITIS (ISUMRY,NHB,3,KSUMRY)
      CALL WRITES (ASUMRY,NHB,3,KSUMRY)
C
      WRITE (NTAPE1) NSB
      IF (NSB .EQ. 0) RETURN
      DO 120 L=1,NSB
      READ (NIT,1001) NOS,ITYPE,JOINT
      ISUMRY(L,1) = NOS
      ISUMRY(L,2) = ITYPE
      ISUMRY(L,3) = JOINT
      IF (IFTSMW(NOS) .NE. NBOD) GO TO 998
      LR = 2*NOS - 1
      JHX = JOINT
      JHY = JHX + NX
      JHZ = JHY + NX
      JSX = JHZ + NX
      JSY = JSX + NX
      JSZ = JSY + NX
      CS(1,LR) = B(JHY,3)
      DS(2,LR) = B(JHZ,1)
      CS(3,LR) = B(JHX,2)
      READ (NIT,1002) (UM2(J),J=1,3)
      ASUMRY(L,1) = UM2(1)
      ASUMRY(L,2) = UM2(2)
      ASUMRY(L,3) = UM2(3)
      CALL ROTTR  (3,ITYPE,UM2,RS(1,1,LR),DUM,DUM)
      DO 125 J=1,NE
      JP6 = J + 6
      BC(1,J) = B(JHX,JP6)
      BC(2,J) = B(JHY,JP6)
      BC(3,J) = B(JHZ,JP6)
      BC(4,J) = B(JSX,JP6)
      BC(5,J) = B(JSY,JP6)
  125 BC(6,J) = B(JSZ,JP6)
      WRITE (NTAPE1) NOS
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,3)
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=4,6)
  120 CONTINUE
      WRITE (NOT,3023) NBOD
      CALL WRITIS (ISUMRY,NSB,3,KSUMRY)
      CALL WRITES (ASUMRY,NSB,3,KSUMRY)
C
      RETURN
C
  998 WRITE (NOT,2001)
 2001 FORMAT (1H1,31HTOPOLOGY ERROR, PROGRAM STOPPED)
      STOP
  999 WRITE (NOT,2002)
 2002 FORMAT (1H1,47HMORE THAN NMODOD MODES SELECTED,PROGRAM STOPPED)
      STOP
C
      END
```

```
      SUBROUTINE MSMODL (NBOD)
C                                                      DEBUG #  16
      IMPLICIT REAL*8(A-H,O-Z)
C
              COMMON /NHNS  /
     *        NHPCI( 6), NSPUI( 6)
              COMMON /NUMBRS/
     *      ZRO,ONE,TAU,IRES                                               0 1713
            COMMON /SPECIF/                                                0 1714
     *      BETA(6, 6),BETAD(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),       16 1715
     *      CH(3,35),DS(3,30),IMU(3, 5),NMCA(6, 6),SIGMA(15),             17 1716
     *      NJ,NH,NSFT,NEMD,NBETA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6),   18 1717
     *      CSCC(14),CCNO(14),NO,NBETA,NLAM,NEQ                           19 1718
            COMMON /SUMRY/                                                 0 1719
     *      RSUMRY(15,6),ISUMRY(15,3),KSUMRY                              96 1720
            COMMON /TAPENO/                                                0 1721
     *      NTAPE1,NTAPE2,NTAPE3                                           0 1722
C                                                                          0 1723
      DIMENSION A( 500,12),B( 500,12),C( 500,12),AMJ( 13, 18)             64 1724
      DIMENSION WS( 500,13),UVEC( 500),VV(3)                              65 1725
C                                                                          0 1726
      DATA  KJOINT, KMODE                                                  0 1727
     *  /     500,     12    /                                            66 1728
      DATA NIT,NOT /5,6/                                                   0 1729
C                                                                          0 1730
 1001 FORMAT(16I5)                                                         0 1731
 1002 FORMAT(8E10.3)                                                       0 1732
 3022 FORMAT (//5X 5HFOR BODY I3,29H THE P-Q HINGE NO., THE EULER          0 1733
     * 37H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE P-Q,/      0 1734
     *  5X54H HINGE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS         0 1735
     *      43HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/         0 1736
     * 5X44H POSITION THE HINGE TRIAD WRT THE BODY TRIAD)                  0 1737
 3023 FORMAT (//5X 5HFOR BODY I3,32H THE SENSOR POINT NO., THE EULER       0 1738
     *40H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE SENSOR,/    0 1739
     *5X53HPOINT APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS            0 1740
     *      43HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/         0 1741
     * 5X45H POSITION THE SENSOR TRIAD WRT THE BODY TRIAD)                 0 1742
C                                                                          0 1743
      KA   = KJOINT                                                        0 1744
      KB   = KJOINT                                                        0 1745
      KC   = KJOINT                                                        0 1746
      KWS  = KJOINT                                                        0 1747
      KAMJ = KMODE + 6                                                     0 1748
C                                                                          0 1749
C***  INITIALIZE NTAPE2, NTAPE3                                            0 1750
      REWIND NTAPE2                                                        0 1751
      REWIND NTAPE3                                                        0 1752
C                                                                          0 1753
C**********************************************                            0 1754
C                                                                          0 1755
C***  NJ = NUMBER OF MASS POINTS ON BODY I                                 0 1756
C***  NE = NUMBER OF ELASTIC MODES RETAINED FOR BODY I                     0 1757
      NE  = IRGFLX(NBOD)                                                   0 1758
      NE6 = NE + 6                                                         0 1759
C                                                                          0 1760
C***  READ FORMA TAPE OR CARDS - CREATE NTAPE2                             0 1761
C                                                                          0 1762
C     MASSES                                                               0 1763
      CALL READ (A,N1,N2,KJOINT,KMODE)                                     0 1764
      NJ = N1                                                              0 1765
      DO 2 I=1,NJ                                                          0 1766
      IF(A(I,1) .LT. ZRO) GO TO 995                                        0 1767
    2 CONTINUE                                                             0 1768
      WRITE(NTAPE2) (A(I,1),I=1,NJ)                                        0 1769
C                                                                          0 1770
C     INERTIAS                                                             0 1771
      CALL READ (A,N1,N2,KJOINT,KMODE)                                     0 1772
      IF(N1 .NE. NJ  .OR.  N2 .NE.  6) GO TO 999
      WRITE(NTAPE2) ((A(I,J),J=1,6),I=1,NJ)
C
C     STATIC MOMENTS - GEOMETRIC COORDINATES
      DO 5 K=1,2
      CALL READ (A,N1,N2,KJOINT,KMODE)
      IF(N1 .NE. NJ  .OR.  N2 .NE.  3) GO TO 999
      WRITE(NTAPE2) ((A(I,J),J=1,3),I=1,NJ)
    5 CONTINUE
C
C     MODAL AMPLITUDES
      DO 10 K=1,6
      CALL READ (A,N1,N2,KJOINT,KMODE)
      IF(N1 .NE. NJ  .OR.  N2 .NE. NE) GO TO 999
      WRITE(NTAPE2) ((A(I,J),J=1,NE),I=1,NJ)
   10 CONTINUE
C
C     STIFFNESS - DAMPING
      DO 20 K=1,2
      CALL READ (A,N1,N2,KJOINT,KMODE)
      IF(N1 .NE. NE  .OR.  N2 .NE. NE) GO TO 999
      WRITE(NTAPE2) ((A(I,J),J=1,NE),I=1,NE)
   20 CONTINUE
C
      DO 47 I=1,NJ
   47 UVEC(I) = ONE
C
      REWIND NTAPE2
      NREC2 = 0
C
C***  CREATE NTAPE3
      CALL CRET3 (NREC2,NJ,NE,A,B,WS,KA,KB,KWS)
C
      REWIND NTAPE3
      NREC3 = 0
C
C***  FORM MUZERO MATRIX
      CALL ZERO(AMU,NE6,NE6,KAMU)
C
      CALL CREMUO(NREC3,NJ,UVEC,A,WS,AMU,KA,KWS,KAMU)
C
C***  FORM AO AND DO COEFFICIENTS
      CALL CREADO(NREC3,NJ,NE,UVEC,A,B,C,WS,AMU,KA,KB,KC,KWS,KAMU)
C
C***  FORM E COEFFICIENTS
      CALL CREE  (NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)
C
C***  SYMMETRIZE AMU
      DO 48 I=1,NE6
      DO 48 J=1,NE6
   48 AMU(J,I) = AMU(I,J)
C
      CALL WRITE(AMU,NE6,NE6,3HMUO,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE6),I=1,NE6)
C
C***  FORM ACOF
      CALL CREA  (NREC3,NJ,NE,UVEC,A,B,KA,KB,KWS)
C
C***  FORM BCOF
      CALL CREB  (NREC3,NREC2,NJ,NE,A,B,WS,KA,KB,KWS)
```

```
C
C***  FORM COEF
      CALL CREC  (NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)
C
C***  FETCH AND STORE STIFFNESS AND DAMPING
      CALL FETCH(NTAPE2,11,NREC2,AMU,NE,NE,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
      CALL FETCH(NTAPE2,12,NREC2,AMU,NE,NE,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  READ AND STORE INITIAL CONDITIONS
      READ(NIT,1002) (A(1,J),J=1,NE)
      READ(NIT,1002) (A(2,J),J=1,NE)
      CALL WRITE(A(1,1),1,NE,4HXEO ,KA)
      CALL WRITE(A(2,1),1,NE,4HXEDO,KA)
      WRITE(NTAPE1) (A(1,J),J=1,NE)
      WRITE(NTAPE1) (A(2,J),J=1,NE)
C
C***  HINGE LOOP  ********************************************
C
      NHB = NFFOI(NBOD)
      WRITE(NTAPE1) NHB
C
      DO 150 L=1,NHB
      READ(NIT,1001) NOH,ITYPE,JOINT
      ISUMRY(L,1) = NOH
      ISUMRY(L,2) = ITYPE
      ISUMRY(L,3) = JOINT
      IF(NOH .LT. 2  .OR.  NOH .GT. NH) GO TO 995
      LR = 6*(NOH-2) + 1
      LD = 7*(NOH-2) + 1
      IF(ITOPLE(1,NOH) .EQ. NBOD) GO TO 152
      IF(ITOPLE(2,NOH) .NE. NBOD) GO TO 996
      LR = LR+1
      LD = LD+1
  152 CONTINUE
C
      GM(1,LD) = WS(JOINT,11)
      GM(2,LD) = WS(JOINT,12)
      GM(3,LD) = WS(JOINT,13)
C
C***  READ ANGLES
      READ(NIT,1002) (WV(J),J=1,3)
      ASUMRY(L,1) =  WV(1)
      ASUMRY(L,2) =  WV(2)
      ASUMRY(L,3) =  WV(3)
C
      CALL ROTTR (3,ITYPE,WV,RH(1,1,LR),DUM,DUM)
C
C***  READ HX,HY,HZ
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL FETCH(NTAPE2, 7,NREC2,C,NJ,NE,KC)
C
      DO 154 J=1,NE
      AMU(1,J) = A(JOINT,J)
      AMU(2,J) = B(JOINT,J)
  154 AMU(3,J) = C(JOINT,J)
C
C***  READ SIGX,SIGY,SIGZ
```

```
      CALL FETCH(NTAPE2, 8,NREC2,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 9,NREC2,B,NJ,NE,KB)
      CALL FETCH(NTAPE2,10,NREC2,C,NJ,NE,KC)
C
      DO 155 J=1,NE
      AMU(4,J) = A(JOINT,J)
      AMU(5,J) = B(JOINT,J)
  155 AMU(6,J) = C(JOINT,J)
C
      WRITE(NTAPE1) NOH
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,3)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=4,6)
C
  150 CONTINUE
      WRITE (NOT,3022) NBOD
      CALL WRITIS (ISUMRY,NHB,3,KSUMRY)
      CALL WRITES (ASUMRY,NHB,3,KSUMRY)
C
C***  SENSOR LOOP  ****************************************
C
      NSB = NSPOT(NBOD)
      WRITE(NTAPE1) NSB
      IF(NSB .EQ. 0) RETURN
C
      DO 160 L=1,NSB
      READ(NIT,1001) NOS,ITYPE,JOINT
      ISUMRY(L,1) = NOS
      ISUMRY(L,2) = ITYPE
      ISUMRY(L,3) = JOINT
      IF(IPTSMW(NOS) .NE. NBOD) GO TO 996
      LR = 2*NOS - 1
      DS(1,LR) = WS(JOINT,11)
      DS(2,LR) = WS(JOINT,12)
      DS(3,LR) = WS(JOINT,13)
C
C***  READ ANGLES
      READ(NIT,1002) (WV(J),J=1,3)
      ASUMRY(L,1) =  WV(1)
      ASUMRY(L,2) =  WV(2)
      ASUMRY(L,3) =  WV(3)
C
      CALL ROTTR (3,ITYPE,WV,RS(1,1,LR),DUM,DUM)
C
C***  READ FX,FY,FZ
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL FETCH(NTAPE2, 7,NREC2,C,NJ,NE,KC)
C
      DO 164 J=1,NE
      AMU(1,J) = A(JOINT,J)
      AMU(2,J) = B(JOINT,J)
  164 AMU(3,J) = C(JOINT,J)
C
C***  READ SIGX,SIGY,SIGZ
      CALL FETCH(NTAPE2, 8,NREC2,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 9,NREC2,B,NJ,NE,KB)
      CALL FETCH(NTAPE2,10,NREC2,C,NJ,NE,KC)
C
      DO 165 J=1,NE
      AMU(4,J) = A(JOINT,J)
      AMC(I,J) = B(JOINT,J)
  105 AMC(K,J) = C(JOINT,J)
C
      WRITE(NTAPE1) NOD
      WRITE(NTAPE1) ((AMC(I,J),J=1,NE),I=1,3)
      WRITE(NTAPE1) ((AMC(I,J),J=1,NE),I=4,6)
C
  100 CONTINUE
      WRITE (NOT,3023) NMOD
      CALL WRITES (ISUMRY,NSU,3,KSUMRY)
      CALL WRITES (ASUMRY,NSD,3,KSUMRY)
C
      RETURN
C
  997 WRITE(NOT,2001)
 2001 FORMAT(1H1,46HNEGATIVE OR ZERO LUMPED MASS, PROGRAM STOPPED)
      STOP
  998 WRITE(NOT,2003)
 2003 FORMAT(1H1,31HTOPOLOGY ERROR, PROGRAM STOPPED)
      STOP
  999 WRITE(NOT,2004)
 2004 FORMAT(1H1,41HERROR IN INPUT TO MSMODL, PROGRAM STOPPED)
      STOP
C
      END
```

```
      SUBROUTINE CHCT3(NREC2,NJ,NE,A,B,WS,KA,KB,KWS)
C
C                                                      DEBUG #   17
      IMPLICIT REAL*8(A-H,O-Z)
C
C        LET:
C
C              N = JOINT NUMBER  N=1,....,NJ
C           W(N) = JOINT MASS
C
C       JXX(N) -JXY(N) -JXZ(N)
C      -JXY(N)  JYY(N) -JYZ(N) = JOINT INERTIA TENSOR
C      -JXZ(N) -JYZ(N)  JZZ(N)
C
C         0       SZ(N)  -SY(N)
C       -SZ(N)      0      SX(N) = JOINT MASS MOMENT TENSOR
C        SY(N)  -SX(N)      0
C
C         P1(N) = JOINT MODAL X DISPLACEMENT COMPONENTS
C         P2(N) = JOINT MODAL Y DISPLACEMENT COMPONENTS
C         P3(N) = JOINT MODAL Z DISPLACEMENT COMPONENTS
C         P4(N) = JOINT MODAL X ROTATION COMPONENTS
C         P5(N) = JOINT MODAL Y ROTATION COMPONENTS
C         P6(N) = JOINT MODAL Z ROTATION COMPONENTS
C          X(N) = JOINT X LOCATION
C          Y(N) = JOINT Y LOCATION
C          Z(N) = JOINT Z LOCATION
C
C        THEN:
```

```
C                          ON NTAPE3 RECORD  1   MATRIX    M * P1               0 2007
C                          ON NTAPE3 RECORD  2   MATRIX   SY * P1               0 2008
C                          ON NTAPE3 RECORD  3   MATRIX   SZ * P1               0 2009
C                          ON NTAPE3 RECORD  4   MATRIX    M * P2               0 2010
C                          ON NTAPE3 RECORD  5   MATRIX   SX * P2               0 2011
C                          ON NTAPE3 RECORD  6   MATRIX   SZ * P2               0 2012
C                          ON NTAPE3 RECORD  7   MATRIX    M * P3               0 2013
C                          ON NTAPE3 RECORD  8   MATRIX   SX * P3               0 2014
C                          ON NTAPE3 RECORD  9   MATRIX   SY * P3               0 2015
C                          ON NTAPE3 RECORD 10   MATRIX  JXX * P4               0 2016
C                          ON NTAPE3 RECORD 11   MATRIX  JXY * P4               0 2017
C                          ON NTAPE3 RECORD 12   MATRIX  JXZ * P4               0 2018
C                          ON NTAPE3 RECORD 13   MATRIX   SY * P4               0 2019
C                          ON NTAPE3 RECORD 14   MATRIX   SZ * P4               0 2020
C                          ON NTAPE3 RECORD 15   MATRIX  JXY * P5               0 2021
C                          ON NTAPE3 RECORD 16   MATRIX  JYY * P5               0 2022
C                          ON NTAPE3 RECORD 17   MATRIX  JYZ * P5               0 2023
C                          ON NTAPE3 RECORD 18   MATRIX   SX * P5               0 2024
C                          ON NTAPE3 RECORD 19   MATRIX   SZ * P5               0 2025
C                          ON NTAPE3 RECORD 20   MATRIX  JXZ * P6               0 2026
C                          ON NTAPE3 RECORD 21   MATRIX  JYZ * P6               0 2027
C                          ON NTAPE3 RECORD 22   MATRIX  JZZ * P6               0 2028
C                          ON NTAPE3 RECORD 23   MATRIX   SX * P6               0 2029
C                          ON NTAPE3 RECORD 24   MATRIX   SY * P6               0 2030
C                          ON NTAPE3 RECORD 25   MATRIX    M * X                0 2031
C                          ON NTAPE3 RECORD 26   MATRIX    M * Y                0 2032
C                          ON NTAPE3 RECORD 27   MATRIX    M * Z                0 2033
C                                                                               0 2034
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                          0 2035
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                                      0 2036
C                                                                               0 2037
      DIMENSION A(KA,1),B(KB,1),WS(KWS,1)                                       0 2038
      LOGICAL DEBUG,LEQU                                                        0 2039
      COMMON /LODBUG/ DEBUG(120)                                                0 2040
      EQUIVALENCE (LEQU,DEBUG(17))                                              0 2041
      DATA NOT/6/                                                               0 2042
      IF(LEQU) WRITE(NOT,1000)                                                  0 2043
 1000 FORMAT (' SUBROUTINE CALTS')                                              0 2044
C                                                                               0 2045
C***  FETCH M                                                                   0 2046
      CALL FETCH(NTAPE2, 1,NREC2,WS(1, 1),NJ,1,KWS)                             0 2047
C***  FETCH JXX........JYZ                                                      0 2048
      CALL FETCH(NTAPE2, 2,NREC2,WS(1, 2),NJ,6,KWS)                             0 2049
C***  FETCH SX,SY,SZ                                                            0 2050
      CALL FETCH(NTAPE2, 3,NREC2,WS(1, 8),NJ,3,KWS)                             0 2051
C***  FETCH GEOMETRY                                                            0 2052
      CALL FETCH(NTAPE2, 4,NREC2,WS(1,11),NJ,3,KWS)                             0 2053
C***  FETCH HX                                                                  0 2054
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)                                    0 2055
      CALL STORE(NTAPE3,WS(1, 1),A,B,NJ,NE,KWS,KA,KB)                           0 2056
      IF(LEQU) CALL WRITE(B,NJ,NE,6H  M*P1,KB)                                  0 2057
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)                           0 2058
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SY*P1,KB)                                  0 2059
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)                           0 2060
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SZ*P1,KB)                                  0 2061
C***  FETCH HY                                                                  0 2062
      CALL FETCH(NTAPE2, 6,NREC2,A,NJ,NE,KA)                                    0 2063
      CALL STORE(NTAPE3,WS(1, 1),A,B,NJ,NE,KWS,KA,KB)                           0 2064
      IF(LEQU) CALL WRITE(B,NJ,NE,6H  M*P2,KB)                                  0 2065
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)                           0 2066
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SX*P2,KB)
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SZ*P2,KB)
C***  FETCH HZ
      CALL FETCH(NTAPE2, 7,NREC2,A,NJ,NE,KA)
      CALL STORE(NTAPE3,WS(1, 1),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H  M*P3,KB)
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SX*P3,KB)
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SY*P3,KB)
C***  FETCH SIGX
      CALL FETCH(NTAPE2, 8,NREC2,A,NJ,NE,KA)
      CALL STORE(NTAPE3,WS(1, 2),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJXX*P4,KB)
      CALL STORE(NTAPE3,WS(1, 5),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJXY*P4,KB)
      CALL STORE(NTAPE3,WS(1, 6),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJXZ*P4,KB)
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SY*P4,KB)
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SZ*P4,KB)
C***  FETCH SIGY
      CALL FETCH(NTAPE2, 9,NREC2,A,NJ,NE,KA)
      CALL STORE(NTAPE3,WS(1, 5),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJXY*P5,KB)
      CALL STORE(NTAPE3,WS(1, 3),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJYY*P5,KB)
      CALL STORE(NTAPE3,WS(1, 7),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJYZ*P5,KB)
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SX*P5,KB)
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SZ*P5,KB)
C***  FETCH SIGZ
      CALL FETCH(NTAPE2,10,NREC2,A,NJ,NE,KA)
      CALL STORE(NTAPE3,WS(1, 6),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJXZ*P6,KB)
      CALL STORE(NTAPE3,WS(1, 7),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJYZ*P6,KB)
      CALL STORE(NTAPE3,WS(1, 4),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6HJZZ*P6,KB)
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SX*P6,KB)
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)
      IF(LEQU) CALL WRITE(B,NJ,NE,6H SY*P6,KB)
      CALL STORE(NTAPE3,WS(1, 1),WS(1,11),B,NJ,1,KWS,KWS,KB)
      IF(LEQU) CALL WRITE(B,NJ,1,6H   M*X ,KB)
      CALL STORE(NTAPE3,WS(1, 1),WS(1,12),B,NJ,1,KWS,KWS,KB)
      IF(LEQU) CALL WRITE(B,NJ,1,6H   M*Y ,KB)
      CALL STORE(NTAPE3,WS(1, 1),WS(1,13),B,NJ,1,KWS,KWS,KB)
      IF(LEQU) CALL WRITE(B,NJ,1,6H   M*Z ,KB)
C
      RETURN
      END
      SUBROUTINE CREMUO(NREC3,NJ,UVEC,A,WS,AMU,KA,KWS,KAMU)
C
C                                                      DEBUG #  18
      IMPLICIT REAL*8(A-H,O-Z)
C
C         USE THE LUMPED PARAMETER DATA ON NTAPE3,  ALONG WITH EQUATIONS
C         II-30,31,35 AND 37 TO COMPUTE THE INERTIA TENSOR, STATIC MASS
C         MOMENT AND TOTAL MASS FOR THE UNDEFORMED FLEXIBLE BODY.
C         THESE TERMS ARE STORED AS A 6 BY 6 MATRIX AND DEFINE THE
C         UPPER LEFT 2 BY 2 PARTITION OF EQUATION II-87
C
C
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3
C
      DIMENSION A(KA,1),WS(KWS,1),AMU(KAMU,1),UVEC(1)
C
C***  FETCH M*RHOX,M*RHOY,M*RHOZ
      CALL FETCH(NTAPE3,25,NREC3,A(1,1),NJ,1,KA)
      CALL FETCH(NTAPE3,26,NREC3,A(1,2),NJ,1,KA)
      CALL FETCH(NTAPE3,27,NREC3,A(1,3),NJ,1,KA)
C***  FORM JXXO
      CALL SATB( ONE,WS(1,13), A(1, 3),AMU(1,1),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB( ONE,WS(1,12), A(1, 2),AMU(1,1),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB( TWO,WS(1,13),WS(1,10),AMU(1,1),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( TWO,WS(1,12),WS(1, 9),AMU(1,1),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( ONE,UVEC    ,WS(1, 2),AMU(1,1),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM JXYO
      CALL SATB(-ONE,WS(1,12), A(1, 1),AMU(1,2),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB(-ONE,WS(1,12),WS(1, 8),AMU(1,2),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB(-ONE,WS(1,11),WS(1, 9),AMU(1,2),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB(-ONE,UVEC    ,WS(1, 5),AMU(1,2),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM JXZO
      CALL SATB(-ONE,WS(1,13), A(1, 1),AMU(1,3),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB(-ONE,WS(1,13),WS(1, 8),AMU(1,3),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB(-ONE,WS(1,11),WS(1,10),AMU(1,3),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB(-ONE,UVEC    ,WS(1, 6),AMU(1,3),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM JYYO
      CALL SATB( ONE,WS(1,13), A(1, 3),AMU(2,2),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB( ONE,WS(1,11), A(1, 1),AMU(2,2),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB( TWO,WS(1,13),WS(1,10),AMU(2,2),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( TWO,WS(1,11),WS(1, 8),AMU(2,2),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( ONE,UVEC    ,WS(1, 3),AMU(2,2),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM JYZO
      CALL SATB(-ONE,WS(1,13), A(1, 2),AMU(2,3),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB(-ONE,WS(1,13),WS(1, 9),AMU(2,3),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB(-ONE,WS(1,12),WS(1,10),AMU(2,3),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB(-ONE,UVEC    ,WS(1, 7),AMU(2,3),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM JZZO
      CALL SATB( ONE,WS(1,12), A(1, 2),AMU(3,3),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB( ONE,WS(1,11), A(1, 1),AMU(3,3),NJ, 1, 1,KWS, KA,KAMU)
      CALL SATB( TWO,WS(1,12),WS(1, 9),AMU(3,3),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( TWO,WS(1,11),WS(1, 8),AMU(3,3),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( ONE,UVEC    ,WS(1, 4),AMU(3,3),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM SXO
      CALL SATB( ONE,UVEC    ,WS(1, 8),AMU(3,5),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( ONE,WS(1,11),WS(1, 1),AMU(3,5),NJ, 1, 1,KWS,KWS,KAMU)
C***  FORM SYO
      CALL SATB( ONE,UVEC    ,WS(1, 9),AMU(1,6),NJ, 1, 1,KWS,KWS,KAMU)
      CALL SATB( ONE,WS(1,12),WS(1, 1),AMU(1,6),NJ, 1, 1,KWS,KWS,KAMU)
```

```
C***  FORM S20                                                             0 2183
      CALL SATB( ONE,UVEC     ,WS(1,17),AMU(2,4),NJ, 1, 1,KWS,KWS,KAMU)    0 2184
      CALL SATB( ONE,WS(1,13),WS(1, 1),AMU(2,4),NJ, 1, 1,KWS,KWS,KAMU)    0 2185
      AMU(2,3) = -AMU(3,2)                                                 0 2186
      AMU(3,1) = -AMU(1,3)                                                 0 2187
      AMU(1,2) = -AMU(2,1)                                                 0 2188
C***  FORM W2S                                                             0 2189
      CALL SATB( ONE,UVEC     ,WS(1, 1),AMU(4,4),NJ, 1, 1,KWS,KWS,KAMU)    0 2190
      AMU(5,5) = AMU(4,4)                                                  0 2191
      AMU(6,6) = AMU(4,4)                                                  0 2192
C                                                                          0 2193
      CALL WRITE(AMU(1,1),3,3,4HINER,KAMU)                                 0 2194
      CALL WRITE(AMU(1,4),3,3,4HSTAT,KAMU)                                 0 2195
      CALL WRITE(AMU(4,4),3,3,4HMASS,KAMU)                                 0 2196
C                                                                          0 2197
      RETURN                                                               0 2198
      END                                                                  0 2199
```

```
      SUBROUTINE CHWAJJ(NRECS,NJ,NE,UVEC,A,B,C,WS,AMU,KA,KB,KC,KWS,KAMU)   0 2200
C                                                                          0 2201
C                                                        DEBUG #  19       0 2202
C                                                                          0 2203
C        USE THE LUMPED PARAMETER DATA ON NTAPE3,  ALONG WITH EQUATIONS   0 2204
C        II-23,30 TO COMPUTE THE 3 BY IDOFLX(N) MATRIX OF D-COEFFICIENT   0 2205
C        AND THE 3 BY IDOFLX(N) MATRIX OF A-COEFFICIENTS IN EQ II-87      0 2206
C        THESE ARE INERTIAL COUPLING COEFFICIENTS WHICH ARE INDEPENDENT   0 2207
C        OF DEFORMATION                                                    0 2208
C                                                                          0 2209
      IMPLICIT REAL*8(A-H,O-Z)                                             0 2210
C                                                                          0 2211
      COMMON /NUMBRS/ ZRO,ONE,TWO,THRS                                     0 2212
      COMMON /TAPES/ NTAPE1,NTAPE2,NTAPE3                                  0 2213
C                                                                          0 2214
      DIMENSION A(KA,1),B(KB,1),C(KC,1),WS(KWS,1),AMU(KAMU,1),UVEC(1)      0 2215
C                                                                          0 2216
C***  FETCH M*RX,M*RY,M*RZ                                                 0 2217
      CALL FETCH(NTAPE3, 1,NRECS,A,NJ,NE,KA)                               0 2218
      CALL FETCH(NTAPE3, 4,NRECS,B,NJ,NE,KB)                               0 2219
      CALL FETCH(NTAPE3, 7,NRECS,C,NJ,NE,KC)                               0 2220
      CALL SATB( ONE,UVEC     ,A,AMU(4,7),NJ, 1,NE,KWS,KA,KAMU)            0 2221
      CALL SATB( ONE,UVEC     ,B,AMU(5,7),NJ, 1,NE,KWS,KB,KAMU)            0 2222
      CALL SATB( ONE,UVEC     ,C,AMU(6,7),NJ, 1,NE,KWS,KC,KAMU)            0 2223
      CALL SATB( ONE,WS(1,13),A,AMU(2,7),NJ, 1,NE,KWS,KA,KAMU)             0 2224
      CALL SATB(-ONE,WS(1,12),A,AMU(3,7),NJ, 1,NE,KWS,KA,KAMU)             0 2225
      CALL SATB(-ONE,WS(1,13),B,AMU(1,7),NJ, 1,NE,KWS,KB,KAMU)             0 2226
      CALL SATB( ONE,WS(1,11),B,AMU(3,7),NJ, 1,NE,KWS,KB,KAMU)             0 2227
      CALL SATB( ONE,WS(1,12),C,AMU(1,7),NJ, 1,NE,KWS,KC,KAMU)             0 2228
      CALL SATB(-ONE,WS(1,11),C,AMU(2,7),NJ, 1,NE,KWS,KC,KAMU)             0 2229
C***  FETCH SY*SIGX,SZ*SIGX                                                0 2230
      CALL FETCH(NTAPE3,13,NRECS,A,NJ,NE,KA)                               0 2231
      CALL FETCH(NTAPE3,14,NRECS,B,NJ,NE,KB)                               0 2232
      CALL SATB( ONE,UVEC     ,A,AMU(6,7),NJ, 1,NE,KWS,KA,KAMU)            0 2233
      CALL SATB(-ONE,UVEC     ,B,AMU(5,7),NJ, 1,NE,KWS,KB,KAMU)            0 2234
      CALL SATB( ONE,WS(1,12),A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)             0 2235
      CALL SATB( ONE,WS(1,13),B,AMU(1,7),NJ, 1,NE,KWS,KB,KAMU)             0 2236
      CALL SATB(-ONE,WS(1,11),A,AMU(2,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB(-ONE,WS(1,11),B,AMU(3,7),NJ, 1,NE,K*3,KB,KAMU)
C***  FETCH SX*SIGY,SZ*SIGY
      CALL FETCH(NTAPE3,18,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,19,NREC3,B,NJ,NE,KB)
      CALL SATB(-ONE,OVEC     ,A,AMU(6,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB( ONE,OVEC     ,B,AMU(4,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB(-ONE,WS(1,12),A,AMU(1,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB( ONE,WS(1,13),B,AMU(2,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB( ONE,WS(1,11),A,AMU(2,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB(-ONE,WS(1,12),B,AMU(3,7),NJ, 1,NE,K*3,KB,KAMU)
C***  FETCH SX*SIGZ,SY*SIGZ
      CALL FETCH(NTAPE3,23,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,24,NREC3,B,NJ,NE,KB)
      CALL SATB( ONE,OVEC     ,A,AMU(5,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB(-ONE,OVEC     ,B,AMU(4,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB(-ONE,WS(1,13),A,AMU(1,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB(-ONE,WS(1,13),B,AMU(2,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB( ONE,WS(1,11),A,AMU(3,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB( ONE,WS(1,12),B,AMU(3,7),NJ, 1,NE,K*3,KB,KAMU)
C***  FETCH SY*HX,SZ*HX
      CALL FETCH(NTAPE3, 2,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3, 3,NREC3,B,NJ,NE,KB)
      CALL SATB( ONE,OVEC     ,B,AMU(2,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB(-ONE,OVEC     ,A,AMU(3,7),NJ, 1,NE,K*3,KA,KAMU)
C***  FETCH SX*HY,SZ*HY
      CALL FETCH(NTAPE3, 5,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3, 6,NREC3,B,NJ,NE,KB)
      CALL SATB(-ONE,OVEC     ,B,AMU(1,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB( ONE,OVEC     ,A,AMU(3,7),NJ, 1,NE,K*3,KA,KAMU)
C***  FETCH SX*HZ,SY*HZ
      CALL FETCH(NTAPE3, 8,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3, 9,NREC3,B,NJ,NE,KB)
      CALL SATB( ONE,OVEC     ,B,AMU(1,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB(-ONE,OVEC     ,A,AMU(2,7),NJ, 1,NE,K*3,KA,KAMU)
C***  FETCH JXX*SIGX,JXY*SIGX,JXZ*SIGX
      CALL FETCH(NTAPE3,10,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,11,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,12,NREC3,C,NJ,NE,KC)
      CALL SATB( ONE,OVEC     ,A,AMU(1,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB(-ONE,OVEC     ,B,AMU(2,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB(-ONE,OVEC     ,C,AMU(3,7),NJ, 1,NE,K*3,KC,KAMU)
C***  FETCH JXY*SIGY,JYY*SIGY,JYZ*SIGY
      CALL FETCH(NTAPE3,15,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,16,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,17,NREC3,C,NJ,NE,KC)
      CALL SATB(-ONE,OVEC     ,A,AMU(1,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB( ONE,OVEC     ,B,AMU(2,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB(-ONE,OVEC     ,C,AMU(3,7),NJ, 1,NE,K*3,KC,KAMU)
C***  FETCH JXZ*SIGZ,JYZ*SIGZ,JZZ*SIGZ
      CALL FETCH(NTAPE3,20,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,21,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,22,NREC3,C,NJ,NE,KC)
      CALL SATB(-ONE,OVEC     ,A,AMU(1,7),NJ, 1,NE,K*3,KA,KAMU)
      CALL SATB(-ONE,OVEC     ,B,AMU(2,7),NJ, 1,NE,K*3,KB,KAMU)
      CALL SATB( ONE,OVEC     ,C,AMU(3,7),NJ, 1,NE,K*3,KC,KAMU)
C
      CALL WRITE(AMU(1,7),3,NE,6HBJCOEF,KAMU)
      CALL WRITE(AMU(4,7),3,NE,6HAJCOEF,KAMU)
C
      RETURN
      END
      SUBROUTINE CREE(NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)
C
C                                                       DEBUG #   20
      IMPLICIT REAL*8(A-H,O-Z)
C
C         USE THE LUMPED PARAMETER DATA ON NTAPE3; AND EQ II-34 TO
C         COMPUTE THE IRGFLX(N) BY IRGFLX(N) MODAL MASS MATRIX. THIS
C         IS THE E MATRIX IN EQ II-87
C
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3
C
      DIMENSION A(KA,1),B(KB,1),C(KC,1),AMU(KAMU,1)
C
C***  FETCH M*FX,SZ*SIGY,SY*SIGZ
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,19,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,24,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HX
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)
C***  FETCH M*HY,SX*SIGZ,SZ*SIGX
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,23,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,14,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HY
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)
C***  FETCH M*FZ,SY*SIGX,SX*SIGY
      CALL FETCH(NTAPE3, 7,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,13,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,18,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HZ
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)
C***  FETCH JXX*SIGX,JXY*SIGY,JXZ*SIGZ
      CALL FETCH(NTAPE3,10,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,15,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,20,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A,-ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH SY*HZ,SZ*HY
      CALL FETCH(NTAPE3, 9,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3, 6,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH SIGX
      CALL FETCH(NTAPE2, 8,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)
C***  FETCH JXY*SIGX,JYY*SIGY,JYZ*SIGZ
      CALL FETCH(NTAPE3,11,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,16,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,21,NREC3,C,NJ,NE,KC)
      CALL ADD3(-ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH SZ*HX,SX*HZ
      CALL FETCH(NTAPE3, 3,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3, 6,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH SIGY
      CALL FETCH(NTAPE2, 9,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMJ(7,7),NJ,NE,NE,KB,KA,KAMJ)
C***  FETCH JXZ*SIGX,JYZ*SIGY,JZZ*SIGZ
      CALL FETCH(NTAPE3,12,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,17,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,22,NREC3,C,NJ,NE,KC)
      CALL ADD3(-ONE,A,-ONE,B, ONE,C,NJ,NE,KA)
C***  FETCH SX*HY,SY*HX
      CALL FETCH(NTAPE3, 5,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3, 2,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A,ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH SIGZ
      CALL FETCH(NTAPE2,10,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMJ)
      CALL WRITE(AMU(7,7),NE,NE,SHECOEF,KAMU)
C
      RETURN
      END
```

```
      SUBROUTINE CREA(NREC3,NJ,NE,JVEC,A,B,KA,KB,KWS)
C
C                                                           DEBUG #  21
      IMPLICIT REAL*8(A-H,O-Z)
C
C          USE THE LUMPED PARAMETER DATA ON NTAPE3, AND EQ II-33,35
C          ALONG WITH A-12 TO A-20 TO COMPUTE THE ALPHA COEFFICIENTS OF
C          EQ II-86.    USED TO DEFINE THE LINEAR DEPENDENCE OF BODY'S
C          STATIC MASS MOMENT ON DEFORMATION
C
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3
C
      DIMENSION A(KA,1),B(KB,1),JVEC(1)
C
      CALL ZERO(A,9,NE,KA)
C***  FETCH M*HX
      CALL FETCH(NTAPE3, 1,NREC3,B,NJ,NE,KB)
      CALL SATB(-ONE,JVEC,B,A(6,1),NJ,1,NE,KWS,KB,KA)
C***  FETCH M*HY
      CALL FETCH(NTAPE3, 4,NREC3,B,NJ,NE,KB)
      CALL SATB( ONE,JVEC,B,A(3,1),NJ,1,NE,KWS,KB,KA)
C***  FETCH M*HZ
      CALL FETCH(NTAPE3, 7,NREC3,B,NJ,NE,KB)
      CALL SATB(-ONE,JVEC,B,A(2,1),NJ,1,NE,KWS,KB,KA)
      DO 45 J=1,NE
      A(4,J) = -A(2,J)
      A(7,J) = -A(3,J)
```

```
   45 A(I,J) = -A(I,J)
C
      CALL WRITE(A,9,NE,6HAACOF,KA)
      WRITE(NTAPE1) ((A(I,J),J=1,NE),I=1,9)
C
      RETURN
      END
```

```
      SUBROUTINE CHB(NREC3,NREC2,NJ,NE,A,B,WS,KA,KB,KWS)
C
C                                                          DEBUG #  22
      IMPLICIT REAL (A-H,O-Z)
C
C          USE THE LUMPED PARAMETER DATA ON NTAPE3 AND EQS II-31,32,37
C          ALONG WITH A-21 TO A-26 TO COMPUTE THE B COEFFICIENTS OF
C          EQ II-63, USED TO DEFINE THE LINEAR DEPENDENCE OF BODY'S
C          INERTIA TENSOR ON DEFORMATION
C
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3
C
      DIMENSION A(KA,1),B(KB,1),WS(KWS,1)
C
      CALL ZERO(B,6,NE,KB)
C***  FETCH Z*RHOX,Z*RHOY,M*RHOZ
      CALL FETCH(NTAPE3,25,NREC3,A(1,1),NJ,1,KA)
      CALL FETCH(NTAPE3,26,NREC3,A(1,2),NJ,1,KA)
      CALL FETCH(NTAPE3,27,NREC3,A(1,3),NJ,1,KA)
      DO 52 I=1,NJ
      WS(I, 8) = WS(I, 8) + A(I,1)
      WS(I, 9) = WS(I, 9) + A(I,2)
   52 WS(I,10) = WS(I,10) + A(I,3)
C***  FETCH HX
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)
      CALL GATE(ONE,WS(1, 8),A,B(2,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1, 8),A,B(3,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1, 9),A,B(4,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1,10),A,B(5,1),NJ,1,NE,KWS,KA,KB)
C***  FETCH HY
      CALL FETCH(NTAPE2, 6,NREC2,A,NJ,NE,KA)
      CALL GATE(ONE,WS(1, 9),A,B(1,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1, 9),A,B(3,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1, 8),A,B(4,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1,10),A,B(6,1),NJ,1,NE,KWS,KA,KB)
C***  FETCH HZ
      CALL FETCH(NTAPE2, 7,NREC2,A,NJ,NE,KA)
      CALL GATE(ONE,WS(1,10),A,B(1,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1,10),A,B(2,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1, 8),A,B(5,1),NJ,1,NE,KWS,KA,KB)
      CALL GATE(ONE,WS(1, 9),A,B(6,1),NJ,1,NE,KWS,KA,KB)
C
      CALL WRITE(B,6,NE,4HBCOF,KB)
      WRITE(NTAPE1) ((B(I,J),J=1,NE),I=1,6)
C
      RETURN
      END
      SUBROUTINE CREC(NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)
C
C                                                       DEBUG #  23
      IMPLICIT REAL*8(A-H,O-Z)
C
C         USE THE LUMPED PARAMETER DATA ON NTAPE3 AND EQS II-32,36
C         ALONG WITH A-27,28,29 TO COMPUTE THE C COEFFICIENTS OF
C         EQ II-88, USED TO DEFINE THE LINEAR DEPENDENCE OF BODY'S
C         COUPLING COEFFICIENTS ON DEFORMATION.
C         ALSO WITH EQS II-31,32,37 ALONG WITH A-30 TO A-35 COMPUTE THE
C         C COEFFICIENTS OF EQ II-89, USED TO DEFINE THE QUADRATIC
C         DEPENDENCE OF BODY'S INERTIA TENSOR ON DEFORMATION
C
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3
C
      DIMENSION A(KA,1),B(KB,1),C(KC,1),AMU(KAMU,1)
C
C***  FORM COEF1 = CXY
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH 4*HY,SX*SIGZ,SZ*SIGX
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,23,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,14,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HX
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C***  FETCH 4*HX,SZ*SIGY,SY*SIGZ
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,19,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,24,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HY
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL SATB(-ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HCXY,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM COEF2 = CXZ
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH HZ
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C***  FETCH 4*HZ,SY*SIGX,SX*SIGY
      CALL FETCH(NTAPE3, 7,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,13,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,16,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HX
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)
      CALL SATB(-ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
```

```
C
      CALL WRITE(AMU,NE,NE,3HCXZ,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM COEF3 = CYZ
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH HY
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C***  FETCH M*HY,SX*SIGZ,SZ*SIGX
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE3,23,NREC3,B,NJ,NE,KB)
      CALL FETCH(NTAPE3,14,NREC3,C,NJ,NE,KC)
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)
C***  FETCH HZ
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)
      CALL SATB(-ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HCYZ,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM COEF4 = C11
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH M*HY AND HY
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C***  FETCH M*HZ AND HZ
      CALL FETCH(NTAPE3, 7,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HC11,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM COEF5 = C22
      CALL ZERO(AMU,NE,NE,KAMU)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C***  FETCH M*HX AND HX
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HC22,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM COEF6 = C33
      CALL ZERO(AMU,NE,NE,KAMU)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C***  FETCH M*HY AND HY
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HC33,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM COEF7 = C12
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH M*HX
```

```
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HC12,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM CCCF8 = C13
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH H2
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HC13,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
C***  FORM CCCF9 = C23
      CALL ZERO(AMU,NE,NE,KAMU)
C***  FETCH M*HY
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)
C
      CALL WRITE(AMU,NE,NE,3HC23,KAMU)
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)
C
      RETURN
      END
```

```
      SUBROUTINE STORE(NTAPE,A,B,Z,NA,NCB,KRA,KRB,KRZ)
C
C                                                          DEBUG #   24
      IMPLICIT REAL*8(A-H,O-Z)
C
C     MATRIX PRODUCT Z = A*B WITH A = DIAGONAL AND STORED AS VECTOR
C         PRODUCT WRITTEN BY ROWS ON NTAPE
C         WHERE A = INPUT   (NA,NA)
C               B = INPUT   (NA,NCB)
C               Z = OUTPUT  (NA,NCB)
C
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
C
      DO 10 I=1,NA
      S = A(I,1)
      DO 10 J=1,NCB
   10 Z(I,J) = S*B(I,J)
C
      WRITE(NTAPE) ((Z(I,J),J=1,NCB),I=1,NA)
C
      RETURN
      END
      SUBROUTINE FETCH(NTAPE,NRECN,NRECO,A,NRA,NCA,KRA)
C                                                            DEBUG #   25
      IMPLICIT REAL*8(A-H,O-Z)
C
C     FETCH MATRIX FROM NTAPE - ASSUMED WRITTEN BY ROWS
C         WHERE NRECN = INPUT RECORD NUMBER DESIRED
C               NRECO = LAST OUTPUT RECORD NUMBER FETCHED FROM NTAPE
C                       USED TO AVOID EXCESSIVE TAPE SEARCH
C               A     = ARRAY WHERE RECORD STORED (NRA,NCA)
C
      DIMENSION A(KRA,1)
C
      IF(NRECN - NRECO) 1,2,3
C
    1 REWIND NTAPE
      NSKIP = NRECN - 1
      IF(NSKIP .EQ. 0) GO TO 20
      DO 10 K=1,NSKIP
   10 READ(NTAPE) DUM
      GO TO 20
C
    2 BACKSPACE NTAPE
      GO TO 20
C
    3 NSKIP = NRECN - NRECO - 1
      IF(NSKIP .EQ. 0) GO TO 20
      DO 11 K=1,NSKIP
   11 READ(NTAPE) DUM
      GO TO 20
C
   20 NRECO = NRECN
C
      READ(NTAPE) ((A(I,J),J=1,NCA),I=1,NRA)
C
      RETURN
      END
```

```
      SUBROUTINE MATM(S,A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)
C
C                                                            DEBUG #   26
      IMPLICIT REAL*8(A-H,O-Z)
C                                              T
C     MATRIX PRODUCT    Z(NEW) = Z(OLD) + S*A *B
C         WHERE S = SCALAR
C               A = INPUT    (NRA,NCA)
C               B = INPUT    (NRA,NCB)
C               Z = OUTPUT   (NCA,NCB)
C
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
C
      DO 20 I=1,NCA
      DO 20 J=1,NCB
      DO 20 K=1,NRA
   20 Z(I,J) = Z(I,J) + S*A(K,I)*B(K,J)
C
      RETURN
      END
      SUBROUTINE ADD3(ALPHA,A,BETA,B,GAMMA,C,NR,NC,KR)                  0 2677
      IMPLICIT REAL*8(A-H,O-Z)                                          0 2678
C                                                         DEBUG #  27   0 2679
C                                                                       0 2680
C     MATRIX ADDITION  A = ALPHA*A + BETA*B + GAMMA*C                   0 2681
C          WHERE ALPHA, BETA, GAMMA ARE INPUT SCALARS AND               0 2682
C          A, B, C ARE INPUT MATRICES (NR,NC)                           0 2683
C                                                                       0 2684
      DIMENSION A(KR,1),B(KR,1),C(KR,1)                                 0 2685
C                                                                       0 2686
      DO 10 I=1,NR                                                      0 2687
      DO 10 J=1,NC                                                      0 2688
   10 A(I,J) = ALPHA*A(I,J) + BETA*B(I,J) + GAMMA*C(I,J)               0 2689
C                                                                       0 2690
      RETURN                                                            0 2691
      END                                                               0 2692



      SUBROUTINE RRIGID (N)                                             0 2693
C                                                         DEBUG #  28   0 2694
C                                                                       0 2695
C        SEE SUBROUTINE DEF3,DEBUG#114,FOR INPUT DATA DEFINITION        0 2696
      IMPLICIT REAL*8(A-H,O-Z)                                          0 2697
C                                                                       0 2698
            COMMON /NHNS  /                                             0 2699
     *      NHPOI( 6), NSPOI( 6)                                       12 2700
            COMMON /SPECIF/                                             0 2701
     *      DETAH(6, 6),DETAHJ(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),  16 2702
     *      DH(3,30),DS(3,30),IMO(3, 5),NMGA(6, 6),ITSMB(15),          17 2703
     *      NB,NH,NSPT,NCFMO,NDELTA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 2704
     *      LOCC(14),LENG(14),NU,NBETA,NLAM,NEQ                        19 2705
            COMMON /SUMMRY/                                             0 2706
     *      ASUMRY(15,6),ISUMRY(15,3),KSUMRY                           98 2707
            COMMON /TAPENO/                                             0 2708
     *      NTAPE1,NTAPE2,NTAPE3                                        0 2709
C                                                                       0 2710
      DIMENSION V(6),AINER(6,6)                                         0 2711
      DATA NIT,NOT /5, 6/                                               0 2712
 1001 FORMAT (16I5)                                                     0 2713
 1002 FORMAT (8D10.3)                                                   0 2714
 3001 FORMAT (//15X30HSUMMARY OF INPUT DATA FOR BODY,I3,                0 2715
     *   16H WHICH IS RIGID. //3X29HTHE 6X6 INERTIA MATRIX IS ---)      0 2716
 3002 FORMAT (//5X9HFOR BODY I3,53H THE P-Q HINGE NO. AND THE EULER     0 2717
     * 57HROTATION TYPE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH /   0 2718
     *   5X57H IS FOLLOWED BY AN ARRAY CONTAINING EULER ANGLES (1,2,3), 0 2719
     * 58H AND POSITION VECTOR COMPONENTS (4,5,6) THAT POSITION THE /   0 2720
     *   5X30HHINGE TRIAD WRT THE BODY TRIAD)                           0 2721
 3003 FORMAT (//5X9HFOR BODY I3,55H THE SENSOR POINT NO. AND THE EULER  0 2722
     * 5E F ROTATION TYPE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH /
     *    5X53HIS FOLLOWED BY AN ARRAY CONTAINING EULER ANGLES(1,2,3),
     * 56 HAND POSITION VECTOR COMPONENTS (4,5,6) THAT POSITION THE/
     *    5X31HSENSOR TRIAD WRT THE BODY TRIAD)
C
C                                NHB = NUMBER OF  HINGES ON BODY N
C                                NSB = NUMBER OF  SENSOR POINTS ON BODY N
      NHB = NHFOI(N)
      NSB = NSFOI(N)
C
C     CREATE 6X6 INERTIA MATRIX
      CALL ZERO  (AINER,6,6,6)
C
C******************** READ BODY MASS AND CG LOCATION
      CALL READ (V,N1,N2,1,6)
C        COMPUTE RIGID BODY MASS MOMENT ABOUT C.
C        FOR GENERALITY INERTIA TENSOR DEFINED ABOUT REFERENCE POINT
      DO 5 J=2,4
    5 V(J) = -V(1)*V(J)
      CALL SKEWV3 (V(2),AINER(1,4),1,6)
      DO 6 I=4,6
    6 AINER(I,I) = V(1)
C
C******************** READ INERTIA TENSOR OF BODY RELATIVE TO REFERENCE
C                     POINT. IF NOT CG MASS MOMENT NON-ZERO
      CALL READ (V,N1,N2,1,6)
      DO 7 I=1,3
    7 AINER(I,I) = V(I)
      AINER(1,2) = -V(4)
      AINER(1,3) = -V(5)
      AINER(2,3) = -V(6)
      DO 8 I=1,6
      DO 8 J=1,6
    8 AINER(J,I) = AINER(I,J)
      WRITE (NTAPE1) ((AINER(I,J),J=1,6),I=1,6)
      CALL PAGEHD
      WRITE (NOT,3001) N
C     PRINT BODY INERTIA MATRIX FOR BODY N
      CALL WRITES (AINER,6,6,6)
C
C     CYCLE THROUGH HINGES ON BODY N
      DO 10 I=1,NHB
C
C******************** READ HINGE NUMBER AND EULER TYPE
      READ (NIT,1001) NOH,ITYPE
C                               NOH = HINGE NUMBER
C                               ITYPE = TYPE OF EULER ANGLE SEQUENCE
C                                       HINGE TO BODY TRIAD
      ISUMRY(I,1) = NOH
      ISUMRY(I,2) = ITYPE
      IF (NOH .EQ. 1) GO TO 999
      LR = 6*(NOH - 2) + 3
      LO = 7*(NOH - 2) + 3
      IF (ITOPOL(1,NOH) .EQ. N) GO TO 12
      IF (ITOPOL(2,NOH) .NE. N) GO TO 999
      LR = LR + 1
      LO = LO + 1
C
C******************** READ ANGLES TO ORIENT HINGE AND BODY TRIADS
C                     USE TO COMPUTE TRANSFORMATION MATRIX IN RH
```

```
   12 READ (NIT,1002) (V(J),J=1,3)
C
C******************** READ VECTOR POSITION BODY TO HINGE TRIAD
C                     STORE IN DH ARRAY
      READ (NIT,1002) (DH(J,LD),J=1,3)
C
      DO 11 J=1,3
      J1 = J + 3
      ASUMRY(I,J ) = V(J)
   11 ASUMRY(I,J1) = DH(J,LD)
C
C               BRANCH TO ROTTR TO COMPUTE TRANSFORMATION MATRIX
C               HINGE FRAME TO BODY FRAME
C                              RH = ARRAY TO STORE HINGE TO BODY
C                                   TRANSFORMATION MATRICES
      CALL ROTTR  (3,ITYPE,V,RH(1,1,LR),DUM,DUM)
C
   10 CONTINUE
C     PRINT INITIAL HINGE TO BODY FRAME ORIENTATION DATA
      WRITE (NOT,3002) N
      CALL WRITIS (ISUMRY,NHB,2,KSUMRY)
      CALL WRITES (ASUMRY,NHB,6,KSUMRY)
C
      IF (NSB .EQ. 0) RETURN
      DO 20 I=1,NSB
C                              NOS = SENSOR POINT NUMBER
C
      READ (NIT,1001) NOS,ITYPE
      ISUMRY(I,1) = NOS
      ISUMRY(I,2) = ITYPE
      IF (IFTSM(NOS) .NE. N) GO TO 999
      LR = 2*NOS
C
C******************** READ ANGLES TO ORIENT SENSOR AND BODY TRIADS
C                     USE TO COMPUTE TRANSFORMATION MATRIX IN RS
      READ (NIT,1002) (V(J),J=1,3)
C
C******************** READ VECTOR SENSOR POINT TO BODY TRIAD
C                     STORE IN DS ARRAY
      READ (NIT,1002) (DS(J,LR),J=1,3)
C
      DO 21 J=1,3
      J1 = J + 3
      ASUMRY(I,J ) = V(J)
   21 ASUMRY(I,J1) = DS(J,LR)
C
C               BRANCH TO ROTTR TO GET SENSOR POINT TO BODY FRAME
C               TRANSFORMATION MATRIX
C                              RS = ARRAY TO STORE SENSOR POINT TO BODY
C                                   TRANSFORMATION MATRIX
      CALL ROTTR  (3,ITYPE,V,RS(1,1,LR),DUM,DUM)
C
   20 CONTINUE
      WRITE (NOT,3003) N
      CALL WRITIS (ISUMRY,NSB,2,KSUMRY)
      CALL WRITES (ASUMRY,NSB,6,KSUMRY)
      RETURN
C
  999 WRITE (NOT,2001)
 2001 FORMAT (1H1,45HTOPOLOGY ERROR,SUBROUTINE ARIGID, PROGRAM STOPPED)
      STOP
C
      END
      SUBROUTINE ZERO  (Z,NR,NC,KR)
C                                                               DEBUG #  29
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  Z(KR,1)
C
C  GENERATE A MATRIX OF ZEROES.
C  CODED BY  RE WOHLEN.  FEB 1965.
C
C      SUBROUTINE ARGUMENTS
C  Z  = OUTPUT MATRIX GENERATED. SIZE(NR,NC).
C  NR = INPUT  NUMBER OF ROWS IN MATRIX Z.
C  NC = INPUT  NUMBER OF COLS IN MATRIX Z.
C  KR = INPUT  ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.
C
      DO 10 I=1,NR
      DO 10 J=1,NC
   10 Z(I,J) = 0.D 0
      RETURN
      END




      SUBROUTINE REVISE (A,IVEC,JVEC,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)
C
C                                                               DEBUG #  30
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(KRA,1), IVEC(1), JVEC(1), Z(KRZ,1)
      DATA NOT / 0/
C
C  REARRANGE ROWS AND COLUMNS OF MATRIX A TO FORM MATRIX Z.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  CODED BY RF HRUDA.  FEBRUARY 1965.
C  LAST REVISION BY RE WOHLEN.  OCTOBER 1972.
C
C      SUBROUTINE ARGUMENTS
C  A      = INPUT  MATRIX TO BE REARRANGED. SIZE(NRA,NCA).
C  IVEC   = INPUT  VECTOR. SIZE(NRA).
C                  IVEC(I)=ROW POSITION OF A(ROW I) IN Z.
C                  IF IVEC(I) IS PLUS , Z(ROW IVEC(I)) = +A(ROW I).
C                  IF IVEC(I) IS MINUS. Z(ROW IVEC(I)) = -A(ROW I).
C                  IF IVEC(I) IS ZERO , A(ROW I) IS OMITTED IN Z.
C  JVEC   = INPUT  VECTOR. SIZE(NCA).
C                  JVEC(J)=COL POSITION OF A(COL J) IN Z.
C                  IF JVEC(J) IS PLUS , Z(COL JVEC(J)) = +A(COL J).
C                  IF JVEC(J) IS MINUS. Z(COL JVEC(J)) = -A(COL J).
C                  IF JVEC(J) IS ZERO , A(COL J) IS OMITTED IN Z.
C  Z      = OUTPUT RESULT MATRIX. SIZE(NRZ,NCZ).
C  NRA    = INPUT  NUMBER OF ROWS IN MATRIX A.
```

```
C   NCA    = INPUT  NUMBER OF COLS IN MATRIX A.
C   NRZ    = INPUT  NUMBER OF ROWS IN MATRIX Z.
C   NCZ    = INPUT  NUMBER OF COLS IN MATRIX Z.
C   KRA    = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C   KRZ    = INPUT  ROW DIMENSION OF Z IN CALLING PROGRAM.
C
      DO 10 I=1,NRZ
      DO 10 J=1,NCZ
   10 Z(I,J) = 0.D 0
C
      DO 30 IA=1,NRA
      IZ = IABS(IVEC(IA))
      IF (IZ .EQ. 0)   GO TO 30
C
      IF (IZ .GT. NRZ) GO TO 999
      DO 25 JA=1,NCA
      JZ = IABS(JVEC(JA))
      IF (JZ .EQ. 0)   GO TO 25
C
      IF (JZ .GT. NCZ) GO TO 999
      SIGN = +1.D 0
      IF (IVEC(IA).LT.0 .AND. JVEC(JA).GT.0  .OR.
     *    IVEC(IA).GT.0 .AND. JVEC(JA).LT.0) SIGN=-1.D 0
      Z(IZ,JZ) = SIGN*A(IA,JA)
   25 CONTINUE
   30 CONTINUE
      RETURN
C
  999 WRITE (NOT,1001)
 1001 FORMAT (1H1,32HERROR IN REVISE, PROGRAM STOPPED)
      STOP
      END
```

```
      SUBROUTINE GAUSSI (A,R,N,KAR)
C                                                             DEBUG #   31
C          FIND INVERSE OF MATRIX A RETURN A**(-1) IN R ARRAY
C          SAVE PIVOT ELEMENTS IN DRVEC THESE WILL BE USED TO
C                    COMPUTE DETERMINANT OF A
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(KAR,1), R(KAR,1), IVEC(150)
              COMMON /DRATIO/
     *      IFL1,IFL2,DRVEC(150)
C
      LOGICAL LDBUG,LEQU
      COMMON /LDEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(31))
      DATA EPS,NOT / 1.D-13, 6/
 1001 FORMAT (5X,30HMATRIX SINGULAR IN GAUSSI AT   , I5)
C
      DO 5 I=1,N
      DO 7 J=1,N
    7 R(I,J) = 0.D 0
    5 R(I,I) = 1.D 0
      IF(.NOT.LEQU) GO TO 1
      WRITE(NOT,1002)
      CALL WRITES(A,N,N,KAR)
 1002 FORMAT (' SUBROUTINE GAUSSI, INPUT MATRIX A IS')
    1 CONTINUE
C
      DO 10 L=1,N
      JBIG = 1
      A1 = DABS(A(L,1))
      DO 15 J=2,N
      A2 = DABS(A(L,J))
      IF (A2 .LT. A1) GO TO 15
      A1 = A2
      JBIG = J
   15 CONTINUE
      IVEC(L) = JBIG
      IF (A1 .GT. EPS) GO TO 20
      IF (IFL1 .EQ. 0) GO TO 75
C        IFL1 =1 CALL IS FROM NUMS. ZERO PIVOT ELEMENT IMPLIES
C                 ZERO GAIN FOR NUMERATOR
      IFL2 = 1
      GO TO 100
   75 WRITE (NOT,1001) L
      STOP
   20 CONTINUE
      ALJBIG = A(L,JBIG)
C        STORE ALL PIVOT ELEMENTS IN DRVEC
      DRVEC(L) = ALJBIG
      DO 17 J=1,N
      A(L,J) = A(L,J)/ALJBIG
   17 B(L,J) = B(L,J)/ALJBIG
      DO 25 I=1,N
      AIJBIG = A(I,JBIG)
      IF (I .EQ. L) GO TO 25
      DO 30 J=1,N
      A(I,J) = A(I,J) - AIJBIG*A(L,J)
   30 B(I,J) = B(I,J) - AIJBIG*B(L,J)
   25 CONTINUE
   10 CONTINUE
C
      DO 40 I=1,N
      IK = IVEC(I)
      DO 40 J=1,N
   40 A(IK,J) = B(I,J)
      DO 50 I=1,N
      DO 50 J=1,N
   50 B(I,J) = A(I,J)
C
      SN = 1.D0
      DO 110 L=1,N
      DO 120 J=1,N
      IF (IVEC(J) .EQ. L) GO TO 110
      IF (IVEC(J) .GT. L) SN = -SN
  120 CONTINUE
  110 CONTINUE
      DRVEC(1) = SN*DRVEC(1)
  100 IFL1 = 0
      IF(.NOT.LPGAU) RETURN
      WRITE(NOT,1003)
 1003 FORMAT (' INVERSE OF INPUT MATRIX A IS ')
      CALL WRITES(B,N,N,KAR)
      WRITE(NOT,1004)
 1004 FORMAT (' PIVOT ELEMENTS USED TO GET A**(-1) ARE ')
      CALL WRITES(DRVEC,1,N,1)
      RETURN
      END
```

```
      SUBROUTINE ROTTR (I23,ITYPE,ANG,ROT,RP2,RP3)
C                                                          DEBUG #   32
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ANG(1),ROT(3,3), RP2(3,3), RP3(3,3)
      DIMENSION IP(36),A(3,3,4),IV(3)
            COMMON /NUMBRS/
     *      ZRO,ONE,TWO,TRES
C
C    SUBROUTINE TO COMPUTE 12 TYPES OF EULER ANGLE TRANSFORMATIONS
C    AND/OR PI INVERSE TRANSFORMATIONS.
C    ALSO IF I23 .EQ. 1, COMPUTE RP2 AND RP3 WHICH MULTIPLY
C          D/DT(THETA(2)) AND D/DT(THETA(3)) TO FORM D/DT(PI INVERSE).
C    I23 = (1 IF ROT = PI INVERSE),  = (3 IF ROT = ROT. TRANS.)
C    ITYPE = 1,2,..12 (TYPE OF EULER ANGLE PERMUTATION, SEE IP BELOW)
C    ANG = INPUT EULER ANGLES
C    ROT = EITHER PI INVERSE OR ROT. TRANS.
C
C          SEE VOL 1 APPENDIX B FOR THEORETICAL DEVELOPMENT
C
C          CODED BY CARL BODLEY,  NOV. 20, 1973.
C    ADDITIONS MADE BY CARL BODLEY,  APR. 5, 1974
C
      DATA IP / 1,2,3,    1,2,1,     1,3,1,     1,3,2,
     *          2,3,1,    2,3,2,     2,1,2,     2,1,3,
     *          3,1,2,    3,1,3,     3,2,3,     3,2,1    /,
     *      NOT, EPS / 0, 1.D-06/
C
      LI = 3*(ITYPE - 1)
C
      DO 15 L=1,3
      IV(L) = 1
      DO 10 I=1,3
      DO  5 J=1,3
    5 A(I,J,L) = ZRO
   10 A(I,I,L) = ONE
   15 CONTINUE
C
      LR = 4 - I23
      DO 20 L=LR,3
      K = IP(LI + L)
      S = DSIN(ANG(L))
      C = DCOS(ANG(L))
      GO TO (1,2,3), K
    1 A(2,2,L) =  C
      A(3,3,L) =  C
      A(2,3,L) = -S
      A(3,2,L) =  S
      GO TO 20
    2 A(1,1,L) =  C
      A(3,3,L) =  C
      A(1,3,L) = S                                                      0 3059
      A(3,1,L) = -S                                                     0 3060
      GO TO 40                                                          0 3061
   30 A(1,1,L) = C                                                      0 3062
      A(2,2,L) = C                                                      0 3063
      A(1,2,L) = -S                                                     0 3064
      A(2,1,L) = S                                                      0 3065
   40 CONTINUE                                                          0 3066
      IF (I[illegible] .EQ. 1) GO TO 50                                 0 3067
C                                                                       0 3068
C         [illegible] ROLE OF ROTATION TRANSFORMATION MATRIX NEEDED     0 3069
C         OBTAIN VIA MATRIX PRODUCTS OF ELEMENTARY ROTATION MATRICIES   0 3070
      CALL MULTS (A(1,1,1),A(1,1,2),A(1,1,4),3,3,3,3,3,3)               0 3071
      CALL MULTS (A(1,1,4),A(1,1,3),ROT,3,3,3,3,3,3)                    0 3072
      RETURN                                                            0 3073
C                                                                       0 3074
C         LOOP HERE TO GET THE INVERSE VELOCITY TRANSFORMATIONS         0 3075
   50 DO 65 L=1,3                                                       0 3076
      II = IP(L + 1)                                                    0 3077
   55 IV(II) = L                                                        0 3078
      S2 = SIN(AS(L))                                                   0 3079
      C2 = COS(AS(L))                                                   0 3080
      II = FLOAT(ITYPE)/4.2                                             0 3081
      K = ITYPE - 4*II                                                  0 3082
      GO TO (61,62,63,64), K                                            0 3083
   61 IF (CABS(C2) .LT. EPS) GO TO 999                                  0 3084
      AL = C2                                                           0 3085
      AL2 = AL*AL                                                       0 3086
      CC = [illegible]                                                  0 3087
      ALP = [illegible]/AL2                                             0 3088
      CEP = CNL/AL2                                                     0 3089
      GO TO 65                                                          0 3090
   62 IF (CABS(S2) .LT. EPS) GO TO 999                                  0 3091
      AL = S2                                                           0 3092
      AL2 = AL*AL                                                       0 3093
      CC = C2                                                           0 3094
      ALP = -C2/AL2                                                     0 3095
      CEP = CNL/AL2                                                     0 3096
      GO TO 65                                                          0 3097
   63 IF (CABS(S2) .LT. EPS) GO TO 999                                  0 3098
      AL = -S2                                                          0 3099
      AL2 = AL*AL                                                       0 3100
      CC = C2                                                           0 3101
      ALP = C2/AL2                                                      0 3102
      CEP = -CNL/AL2                                                    0 3103
      GO TO 65                                                          0 3104
   64 IF (CABS(C2) .LT. EPS) GO TO 999                                  0 3105
      AL = C2                                                           0 3106
      AL2 = AL*AL                                                       0 3107
      CC = -S2                                                          0 3108
      ALP = S2/AL2                                                      0 3109
      CEP = CNL/AL2                                                     0 3110
C                                                                       0 3111
   65 K = IP(L+3)                                                       0 3112
      GO TO (71,72,73), K                                               0 3113
   71 A(2,2,1) =  A(2,3,3)                                              0 3114
      A(3,3,1) =  A(2,3,3)                                              0 3115
      A(3,2,1) =  A(2,2,3)                                              0 3116
      A(2,3,1) = -A(2,2,3)                                              0 3117
      A(1,1,1) = ZRO                                                    0 3118
      GO TO 75                                                          0 3119
   72 A(1,1,1) =  A(3,1,3)                                              0 3120
      A(3,3,1) =  A(3,1,3)                                              0 3121
      A(1,3,1) =  A(1,1,3)                                              0 3122
      A(3,1,1) = -A(1,1,3)                                              0 3123
      A(2,2,1) = ZRO                                                    0 3124
      GO TO 75                                                          0 3125
   73 A(1,1,1) =  A(1,2,3)                                              0 3126
      A(2,2,1) =  A(1,2,3)                                              0 3127
      A(2,1,1) =  A(1,1,3)                                              0 3128
      A(1,2,1) = -A(1,1,3)                                              0 3129
      A(3,2,1) = ZRO                                                    0 3130
   75 DO 80 I=1,3                                                       0 3131
      I1 = IV(I)                                                        0 3132
      DO 80 J=1,3                                                       0 3133
      A(I1,J,2) = A(I,J,1)                                              0 3134
   80 A(I1,J,4) = A(I,J,3)                                              0 3135
C                                                                       0 3136
      DO 90 J=1,3                                                       0 3137
      ROT(1,J) = A(1,J,4)/AL                                            0 3138
      RP3(1,J) = A(1,J,2)/AL                                            0 3139
      ROT(2,J) = A(2,J,4)                                               0 3140
      RP3(2,J) = A(2,J,2)                                               0 3141
      ROT(3,J) = A(3,J,4) - BL*ROT(1,J)                                 0 3142
      RP3(3,J) = A(3,J,2) - BL*RP3(1,J)                                 0 3143
      RP2(1,J) = ALP*A(1,J,4)                                           0 3144
      RP2(2,J) = ZRO                                                    0 3145
   90 RP2(3,J) = BEP*A(1,J,4)                                           0 3146
C                                                                       0 3147
      RETURN                                                            0 3148
  999 WRITE (NOT,1000) ITYPE,ANG(2)                                     0 3149
 1000 FORMAT (1H1,22HGIMBAL LOCK-- ITYPE = ,I5,8HANGLE = ,D15.8)        0 3150
      STOP                                                              0 3151
      END                                                               0 3152
```

```
      SUBROUTINE UNITY  (Z,N,KR)                                        0 3153
C                                                       DEBUG #   33    0 3154
      IMPLICIT REAL*8(A-H,O-Z)                                          0 3155
      DIMENSION  Z(KR,1)                                                0 3156
C                                                                       0 3157
C  GENERATE A UNITY MATRIX. (ONES ON THE DIAGONAL).                     0 3158
C  CODED BY  RL WOHLEN.  FEB 1965.                                      0 3159
C                                                                       0 3160
C      SUBROUTINE ARGUMENTS                                             0 3161
C  Z  = OUTPUT MATRIX GENERATED. SIZE(N,N).                             0 3162
C  N  = INPUT  SIZE OF MATRIX Z (SQUARE).                               0 3163
C  KR = INPUT  ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.            0 3164
C                                                                       0 3165
      DO 20 I=1,N                                                       0 3166
      DO 10 J=1,N                                                       0 3167
   10 Z(I,J) = 0.D 0                                                    0 3168
   20 Z(I,I) = 1.D 0                                                    0 3169
      RETURN                                                            0 3170
      END                                                               0 3171
      SUBROUTINE MULTA (AZ,B,NRA,NRB,NCB,KAZ,KB)                           0 3172
C                                                         DEBUG #  34      0 3173
      IMPLICIT REAL*8(A-H,O-Z)                                             0 3174
      DIMENSION AZ(KAZ,1), B(KB,1),W(150)                                 70 3175
      DATA NOT / 6/                                                        0 3176
C                                                                          0 3177
C   MATRIX MULTIPLICATION. A * B = Z.                                      0 3178
C   USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.                       0 3179
C   AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE     0 3180
C   LARGER OF A OR Z.                                                      0 3181
C   DEVELOPED BY C S BODLEY.  JANUARY 1965.                                C 3182
C   LAST REVISION BY R F HRUDA.  JUNE 1972.                                0 3183
C                                                                          0 3184
C       SUBROUTINE ARGUMENTS                                               0 3185
C   AZ  = INPUT  MATRIX. SIZE(NRA,NRB).                                    0 3186
C       = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).                             C 3187
C   B   = INPUT  MATRIX. SIZE(NRB,NCB)                                     0 3188
C   NRA = INPUT  NUMBER OF ROWS OF MATRICES A,Z.                           C 3189
C   NRB = INPUT  NUMBER OF ROWS OF MATRIX B. COLS OF MATRIX A. MAX=500.    0 3190
C   NCB = INPUT  NUMBER OF COLS OF MATRICES B,Z.                           0 3191
C   KAZ = INPUT  ROW DIMENSION OF AZ IN CALLING PROGRAM.                   0 3192
C   KB  = INPUT  ROW DIMENSION OF B IN CALLING PROGRAM.                    0 3193
C                                                                          0 3194
      IF (NRB .GT. 150) GO TO 999                                         71 3195
C                                                                          0 3196
      DO 40 I=1,NRA                                                        0 3197
      DO 20 K=1,NRB                                                        0 3198
   20 W(K) = AZ(I,K)                                                       0 3199
      DO 40 J=1,NCB                                                        0 3200
      S = 0.0D0                                                            0 3201
      DO 30 K=1,NRB                                                        0 3202
   30 S = S + W(K)*B(K,J)                                                  0 3203
   40 AZ(I,J) = S                                                          0 3204
      RETURN                                                               0 3205
C                                                                          0 3206
  999 WRITE (NOT,1001)                                                     0 3207
 1001 FORMAT (1H1,31HERROR IN MULTA, PROGRAM STOPPED)                      0 3208
      STOP                                                                 0 3209
      END                                                                  0 3210
```

```
      SUBROUTINE BTABA (AZ,B,NRB,NCB,KAZ,KB)                               C 3211
C                                                         DEBUG #  35      0 3212
      IMPLICIT REAL*8(A-H,O-Z)                                             0 3213
      DIMENSION AZ(KAZ,1), B(KB,1),W(150)                                 72 3214
      DATA NOT / 6/                                                        0 3215
C                                                                          0 3216
C   TRIPLE MATRIX PRODUCT.  B(TRANSPOSE) * A * B = Z.                      0 3217
C   A MUST BE SYMMETRIC TO GET CORRECT ANSWER.                             0 3218
C   Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF.   0 3219
C   USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.                       0 3220
C   AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE     0 3221
C   LARGER OF A OR Z.                                                      0 3222
C   DEVELOPED BY W A BENFIELD.  MAY 1972.                                  0 3223
C   LAST REVISION BY R A PHILIPPUS.  JUNE 1972.                            0 3224
C   MODIFIED FOR USE IN GSFC PROGRAM BY CARL BODLEY. MAY 1974              0 3225
```

```
C
C       SUBROUTINE ARGUMENTS
C   AZ  = INPUT   INNER MATRIX. SIZE(NRB,NRB).
C       = OUTPUT  RESULT MATRIX. SIZE(NCB,NCB).
C   B   = INPUT   OUTER MATRIX. SIZE(NRB,NCB).
C   NRB = INPUT   NUMBER OF ROWS OF MATRIX B. SIZE OF MATRIX A. MAX=150.
C   NCB = INPUT   NUMBER OF COLS OF MATRIX B. SIZE OF MATRIX Z. MAX=150.
C   KAZ = INPUT   ROW DIMENSION OF AZ IN CALLING PROGRAM.
C   KB  = INPUT   ROW DIMENSION OF B IN CALLING PROGRAM.
C
      IF (NRB.GT.150 .OR. NCB.GT.150 .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ)
     *    GO TO 999
C
      DO 20 I=1,NRB
      DO 5 K=1,NRB
    5 W(K) = AZ(I,K)
      DO 20 J=1,NCB
      S = 0.D 0
      DO 10 K=1,NRB
   10 S = S + W(K)*B(K,J)
   20 AZ(I,J) = S
C
      DO 30 J=1,NCB
      DO 25 I=1,J
      W(I) = 0.D 0
      DO 25 K=1,NRB
   25 W(I) = W(I)+B(K,I)*AZ(K,J)
      DO 30 I=1,J
      AZ(I,J) = W(I)
   30 AZ(J,I) = W(I)
      RETURN
C
  999 WRITE (NOT,1001)
 1001 FORMAT (1H1,31HERROR IN BTABA, PROGRAM STOPPED)
      STOP
      END
```

```
      SUBROUTINE PR3 (A,B,C,W,Z,S,NRA,NCA,NCB,NCC,KA,KB,KC,KW,KZ)
C
C                                                       DEBUG #   36
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(KA,1),B(KB,1),C(KC,1),W(KW,1),Z(KZ,1)
CC
C     W = A*B
CC
      DO 10 I=1,NRA
      DO 10 J=1,NCB
      W(I,J) = 0.D 0
      DO 10 K=1,NCA
   10 W(I,J) = W(I,J) + A(I,K)*B(K,J)
CC
C                  T
C     Z = Z + S*C *W
CC
      DO 20 I=1,NCC
      DO 20 J=1,NEQ                                                     0 3280
      DO 20 K=1,NKA                                                     0 3281
   20 Z(I,J) = Z(I,J) + S*C(K,I)*A(K,J)                                 0 3282
C                                                                       0 3283
      RETURN                                                            0 3284
      END                                                               0 3285
```

```
      SUBROUTINE DYN320                                                 C 3286
C                                                     DEBUG #   37      C 3287
      IMPLICIT REAL*8(A-H,O-Z)                                          0 3288
C                                                                       0 3289
            COMMON /AMODW /                                             C 3290
     *      ATU(22,22, 6),B(22,15)                                      1 3291
            COMMON /COMPAR/                                             0 3292
     *      CSTOTA(150)                                                95 3293
            COMMON /DNAUX /                                             0 3294
     *      NAUX                                                        0 3295
            COMMON /HANDS /                                             C 3296
     *      HATH(3,12,15),SIGH(3,12,15),HATS(3,12,15),SIGS(3,12,15)     4 3297
            COMMON /ILINER/                                             0 3298
     *      ILINER                                                      C 3299
            COMMON /INTGRL/                                             0 3300
     *      AI(171, 6),ACOF(9,12, 6),BCOF(6,12, 6),                     5 3301
     *      COF11(12,12, 6),COF22(12,12, 6),COF33(12,12, 6),AK(12,12, 6),  6 3302
     *      COF12(12,12, 6),COF13(12,12, 6),COF23(12,12, 6),AD(12,12, 6),  7 3303
     *      COFXY(12,12, 6),COFXZ(12,12, 6),COFYZ(12,12, 6)             8 3304
            COMMON /MAXNUM/                                             C 3305
     *      NTMAX,NHMAX,NSPMAX,NMWMAX,NMWODD,NMDDDD,K40,KY,KU           0 3306
            COMMON /MISCNJ/                                             C 3307
     *      NCPRNT,NCPLOT                                               0 3308
            COMMON /NUMBRS/                                             0 3309
     *      ZRO,ONE,TWO,THRE                                            C 3310
            COMMON /PLTDTA/                                             0 3311
     *      NPPLOT,NCPLOT                                               0 3312
            COMMON /SPRKTA/                                             C 3313
     *      SRK(250),PRK(4), NT                                        15 3314
            COMMON /SPECIF/                                             0 3315
     *      BETAM(6, 6),BETAMD(6, 6),AMU(2, 5),RM(3,3,30),RS(3,3,30),  16 3316
     *      DM(3,30),DS(3,30),IMU(3, 5),NMCA(6, 6),IFISMA(15),         17 3317
     *      NT,NH,NSPT,NUFMU,NDELTA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 3318
     *      LOCC(14),LEND(14),ND,NBETA,NLAM,NEQ                        19 3319
            COMMON /TAPENO/                                             C 3320
     *      NTAPE1,NTAPE2,NTAPE3                                        0 3321
            COMMON /TIMESS/                                             0 3322
     *      STARTT,DELTAT,T,ENDT,TMST                                   0 3323
            COMMON /VECTOR/                                             0 3324
     *      Y(250),YDT(250)                                            20 3325
            COMMON /VINDEP/                                             0 3326
     *      INDEP(250)                                                 21 3327
C                                                                       C 3328
      DATA NOT / 6/                                                     0 3329
C                                                                       0 3330
 2001 FORMAT (////1X,47HTHE FOLLOWING INTEGER ARRAY (INDEP) PRESCRIBES   0 3331
     *54HINDEPENDENT VARIABLES (1), AND DEPENDENT VARIABLES (0), /1X,   0 3332
     * 101(1H-))                                                        0 3333
```

```
C
      PRK(1) = ONE/TWO
      PRK(2) = ONE - DSQRT(PRK(1))
      PRK(3) = TWO - PRK(2)
      PRK(4) = PRK(1)
C
      KT = 0
      T     = STARTT
      TMST = ZRO
C
      DO 100 I=1,NEQ
  100 QRK(I) = ZRO
C
C
      REWIND NTAPE1
C
      DO 2 I=1,KY
      INDEF(I) = 1
      Y(I) = ZRO
    2 YDT(I) = ZRO
C
      DO 3 N=1,NB
      NXE = IRGFLX(N)
      NP6 = 6 + NXE
C     READ BODY N INERTIA MATRIX
      READ (NTAPE1) ((AMU(I,J,N),J=1,NP6),I=1,NP6)
C        LOAD UNDEFORMED INERTIA MATRIX INTO THE AM ARRAY.
C        HEREAFTER AMU ARRAY WILL CONTAIN THE DEFORMED INERTIA MATRICES
      KNT = 0
      DO 6 I=1,NP6
      DO 6 J=1,NP6
      KNT = KNT + 1
    6 AM(KNT,N) = AMU(I,J,N)
C
C     GO TO 3 IF BODY N A RIGID BODY
      IF (NXE .EQ. 0) GO TO 3
C
C
      LXE   = LOCQ(N+NB)
      LXED = LOCQ(N    ) + 6
      LXEN  = LXE  + NXE - 1
      LXEDN = LXED + NXE - 1
C
C        ACOF  = ALPHA COEFFICIENTS OF EQ II-88 USED TO DEFINE BODY'S
C                STATIC MASS MOMENTS LINEAR DEPENDENCE ON DEFORMATION
C        BCOF  = B COEFFICIENTS OF EQ II-88 USED TO DEFINE BODY INERTIA
C                TENSOR'S LINEAR DEPENDENCE ON DEFORMATION
C       COFXY = C SUB XY COEFFICIENTS OF EQ II-89
C       COFXZ = C SUB XZ COEFFICIENTS OF EQ II-89
C       COFYZ = C SUB ZY COEFFICIENTS OF EQ II-89
C       COF11 = C SUB 11 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP
C       COF22 = C SUB 22 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP
C       COF33 = C SUB 33 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP
C       COF12 = C SUB 12 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP
C       COF13 = C SUB 13 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP
C       COF23 = C SUB 23 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP
      READ (NTAPE1) ((ACOF(I,J,N),J=1,NXE),I=1,3)
      READ (NTAPE1) ((BCOF(I,J,N),J=1,NXE),I=1,6)
      READ (NTAPE1) ((COFXY(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COFXZ(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COFYZ(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COF11(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COF22(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COF33(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COF12(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COF13(I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((COF23(I,J,N),J=1,NXE),I=1,NXE)
C
C          AK(N) MODAL STIFFNESS MATRIX BODY N
C          AD(N) MODAL DAMPING MATRIX BODY N
      READ (NTAPE1) ((AK   (I,J,N),J=1,NXE),I=1,NXE)
      READ (NTAPE1) ((AD   (I,J,N),J=1,NXE),I=1,NXE)
C
C          INITIAL CONDITIONS FOR FLEXIBLE BODY COORDINATES READ FROM
C          TAPE DIRECTLY INTO STATE VECTOR
      READ (NTAPE1) (Y(J),J=LXE ,LXEN )
      READ (NTAPE1) (Y(J),J=LXED,LXEDN)
C
      READ (NTAPE1) NHB
CCC NOTE ---  NHB IS NO. OF P/G HINGES, NOT TO INCLUDE HINGE-1.
CCC           OVERLAY(1,0) MUST BE SURE OF THIS.
      DO 4 L=1,NHB
      READ (NTAPE1) NOH
CCCC NOH NOT TO INCLUDE HINGE 1  ****
      LHS = 0
      IF (ITOPOL(1,NOH) .EQ. N) LHS = 2*NOH - 3
      IF (ITOPOL(2,NOH) .EQ. N) LHS = 2*NOH - 2
      IF (LHS .LE. 0) GO TO 999
      READ (NTAPE1) ((MATH(I,J,LHS),J=1,NXE),I=1,3)
    4 READ (NTAPE1) ((SIGH(I,J,LHS),J=1,NXE),I=1,3)
      READ (NTAPE1) NSB
      IF (NSB .EQ. 0) GO TO 3
      DO 5 L=1,NSB
      READ (NTAPE1) NOS
      READ (NTAPE1) ((MATS(I,J,NOS),J=1,NXE),I=1,3)
    5 READ (NTAPE1) ((SIGS(I,J,NOS),J=1,NXE),I=1,3)
    3 CONTINUE
C
C
C     LOAD INITIAL MOMENTUM WHEEL RATES IN STATE VECTOR
      DO 10 N=1,NB
      LTC = LCCO(N) + IRGFLX(N) + 3
      NMW = NMCW(1,N)
      IF (NMW .EQ. 0) GO TO 10
      NMWVS = 0
      DO 15 I=1,NMW
      NOMW = NMCW(I+2,N)
      IF (IAG(3,NOMW) .EQ. 0) GO TO 15
      NMWVS = NMWVS + 1
      Y(LTC+NMWVS) = AMO(1,NOMW)
   15 CONTINUE
   10 CONTINUE
C
C     LOAD INITIAL HINGE ANGLES AND DISPLACEMENTS IN STATE VECTOR
      LOE = LCCO(2*NB+1) - 1
      DO 20 J=1,NH
      DO 20 I=1,6
      IP1 = I + 1
      IF (IHDATA(IP1,J) .EQ. 1) GO TO 20
      LOE = LOE + 1
      Y(LBE) = BETAH(I,J)
   20 CONTINUE
C
C     LOAD INITIAL VALUES FOR CONTROL SYSTEM VARIABLES IN STATE VECTOR
      IF (NDELTA .EQ. 0) GO TO 25
      LDE  = LOCU(2*NB+2) - 1
      DO 26 J=1,NDELTA
      L = LDE + J
   26 Y(L) = CNTDTA(J)
C
C        BRANCH TO ROTOM TO GET RELATIVE ORIENTATION AND POSITION
C        DATA FOR CONTIGUOUS BODIES AND BODY FIXED FRAMES RELATIVE TO
C        INERTIAL REFERENCE
   25 CALL ROTOM
C        BRANCH TO BHGENR TO COMPUTE HINGE POINT KINEMATICS
      CALL BHGENR
C        BRANCH TO FINDU TO GET INDEPENDENT COMPONENTS OF J VECTOR
C        FOR INSERTION INTO THE STATE VECTOR Y
      CALL FINDU (0)
C
      IPRNT = 0
      IPLOT = 0
C
C        BRANCH TO YDOT TO GET TIME DERIVATIVE OF STATE VECTOR. RESULTS
C        STORED IN YDT ARRAY EQUATIONS II-1 THRU 6 ALL EVALUATED IN YDOT
      CALL YDOT
      WRITE (NOT,2001)
      CALL WRITIS (INDEP,1,NEQ,1)
C
      NCAM = NEQ
      NRAM = NCAM + NAUX
C
C        COMPUTE ENERGY AND MOMENTUM FOR BODIES AND SYSTEM
      CALL ENGMOM
      CALL PRNTOU
      IF (ILINER .EQ. 1) GO TO 50
      CALL PLOTWR
C
C        START INTEGRATION OF EQUATIONS OF MOTION
C             NEQ = TOTAL NUMBER OF EQUATIONS TO INTEGRATE
C        RETURN TO THIS POINT AFTER EACH INTEGRATION CYCLE
   11 CALL RKADAM(NEQ)
      CALL ENGMOM
      IPRNT = IPRNT + 1
      IF (IPRNT .NE. NOPRNT) GO TO 12
      CALL PRNTOU
      IPRNT = 0
   12 CONTINUE
      IPLOT = IPLOT + 1
      IF (IPLOT .NE. NOPLOT) GO TO 13
      CALL PLOTWR
      IPLOT = 0
   13 CONTINUE
      IF (T .LT. ENDT) GO TO 11
      RETURN
C
C        COME HERE IF EQUATIONS TO BE LINEARIZED
   50 CALL LINEAR (NRAM,NCAM)
      RETURN
C
```

```
  295 WRITE (NOT,2010)
 2010 FORMAT (1H1,15HERROR IN ITEPOL)
      STOP
      END
```

```
      SUBROUTINE FINDU (IFLAG)
C                                                       DEBUG #  38
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION ICON(36), IVEC(36),JCON(113),R(33),SIGMAI( 6),DEIGEN( 6)
C
              COMMON /AMCHA /
     *        AMU(2,22, 6),UA(30,113)
              COMMON /BRCSR3/
     *        HH(6,15,11),BB(6,15,15),RJL(3,3, 6),SJL(3, 6)
              COMMON /NUMBRS/
     *        ZRO,ONE,TAU,THRES
              COMMON /SPECIF/
     *        BETAM(6, 6),BETAMU(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),
     *        UH(3,30),JS(3,30),IMU(3, 5),NMCA(6, 3),ITSMA(15),
     *        NJ,NH,NSPT,NUMMG,NDELTA,ITEPOL(7, 6),IRGFLA( 6),IHDATA(7, 6),
     *        LOCL(14),LENGL(14),NU,NBETA,NLAM,NEQ
              COMMON /TIMESS/
     *        STARTT,DELTAT,T,ENDT,TRST
              COMMON /VECTOR/
     *        Y(250),YDT(250)
              COMMON /VINDEP/
     *        INDEP(250)
              COMMON /MAXNUM/
     *        NJMAX,NHMAX,NSPMAX,N4MAX,NMASDU,NMDSSJ,S4J,NY,KU
C
      DATA EPS,NOT / 1.D-30, 6/
C
      LOGICAL DEBUG,LEQU
      COMMON /LDEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(38))
      IF(LEQU) WRITE(NOT,2011) IFLAG
 2011 FORMAT (' SUBROUTINE FINDU(',I1,')')
      IF (IFLAG .EQ. 2) GO TO 100
      IF (IFLAG .EQ. 1) GO TO 110
C
C        INITIAL PASS THROUGH IFLAG=0
C        IFLAG = 0 INITIAL PASS THROUGH GET INDEPENDENT COORDINATES
C                  TO PUT IN STATE VECTOR Y
C        IFLAG = 1 NOT AN INTERMEDIATE STEP THROUGH RUNGE KUTTA
C                  GET DEPENDENT COORDINATES FOR INCLUSION IN Y ARRAY
C        IFLAG = 2 AN INTERMEDIATE STEP THROUGH RUNGE KUTTA
C        USE SAME SET OF INDEPENDENT COORDINATES AS USED IN IFLAG=1
C
C                  IFLAG=0 SETUP
      NCN = 6*NH
      DO 205 L=1,NCN
      IVEC(L) = 0
  205 ICON(L) = 1
      DO 207 J=1,NU
  207 JCON(J) = 0
      DO 210 N=1,NB
      JU = LOCC(N) - 1
C        JRNG > 6 ONLY FOR NON-TOPOLOGICAL TREES. RELATIVE BODY
C                  COORDINATES NEEDED TO AVOID STATIC INDETERMINANCY
      JRNG = 6
      IF (NH .GT. NB) JRNG = 6 + IRGFLX(N)
      DO 210 J=1,JRNG
  210 JCON(JU+J) = 1
      NR = 0
      DO 215 J=1,NH
      DO 215 I=1,6
      IP1 = I + 1
      NR = NR + 1
C        LOAD R ARRAY WITH INITIAL BETAHD ARRAY
      R(NR) = BETAHD(I,J)
C        CHECK FOR FIXED OR RHENOMIC CONSTRAINT CONDITIONS
      IF (IHDATA(IP1,J) .EQ. 1) R(NR) = ZRO
      IF (IHDATA(IP1,J) .EQ. 2) R(NR) = ADT(NR,T)
      BETAHD(I,J) = R(NR)
  215 CONTINUE
C
      GO TO 150
C
C        COME HERE TO GET CONSTRAINT EQUATIONS SETUP
C                                 IFLAG = 1
C        START SEARCH FOR NEW AND POSSIBLY BETTER SET OF
C        INDEPENDENT COORDINATES(ONLY ON LAST PASS THRU RUNGE KUTTA)
  110 DO 220 L=1,NCN
      IVEC(L) = 0
  220 ICON(L) = 0
      DO 221 J=1,NU
  221 INDEF(J) = 1
C        CONSTRAINT INFO INTO ICON
      NR = 0
      DO 225 J=1,NH
      DO 225 I=2,7
      NR = NR + 1
      ICON(NR) = IHDATA(I,J)
  225 CONTINUE
C
C                                 IFLAG = 2
C        USE SAME SET OF INDEPENDENT COORDINATES AS IN IFLAG=1 PASS
  130 DO 230 L=1,NCN
C        LOAD CONSTRAINT DATA INTO R ARRAY
      IF (ICON(L) .EQ. 0) GO TO 230
      IF (ICON(L) .EQ. 1) R(L) = ZRO
      IF (ICON(L) .EQ. 2) R(L) = ADT(L,T)
  230 CONTINUE
C
C        LOAD BW ARRAY WITH ELEMENTS OF BH ARRAY COMPUTED IN BHGENER
C                  BW = COEFFICIENT MATRIX OF EQUATION II-84
  150 DO 310 I=1,NCN
      DO 310 J=1,NU
  310 BW(I,J) = ZRO
      LEQ = 6 + IRGFLX(I)
      JBIGST(I) = 1
      JBIGFN(I) = LEQ
      DO 315 I=1,6
      DO 315 J=1,LEQ
  315 BW(I,J) = BH(I,J,1)
C
      DO 320 L=2,NH
      LQ = 2*L - 2
      LP = LQ + 1
      NOBQ = ITOPOL(1,L)
      NOBP = ITOPOL(2,L)
      LH = 6*(L-1)
      LBQ = LOCQ(NOBQ) - 1
      LBP = LOCQ(NOBP) - 1
      LEQ = 6 + IRGFLX(NOBQ)
      LEP = 6 + IRGFLX(NOBP)
      JBIGST(L) = LBQ+1
      JBIGFN(L) = LBQ+LEQ
      DO 325 I=1,6
      DO 325 J=1,LEQ
  325 BW(I+LH,J+LBQ) = BH(I,J,LQ)
      DO 330 I=1,6
      DO 330 J=1,LEP
  330 BW(I+LH,J+LBP) = BH(I,J,LP)
  320 CONTINUE
C
      IF(.NOT.LEQU) GO TO 70
      WRITE(NOT,1002)
 1002 FORMAT (' THE BW MATRIX IS')
      NH6 = NHMAX*6
      CALL WRITES(BW,NCN,NQ,NH6)
   70 CONTINUE
C         IF IFLAG = 0
C         USE REDUCTION PROCESS TO SOLVE FOR Q AND TO PICK OUT THE
C         INDEPENDENT COORDINATES TO BE INSERTED INTO THE STATE VECTOR Y
C         IFLAG = 1 OR 2 DEPENDENT COORDINATES IN Y ARRAY COMPUTED
C         IFLAG = 1 CHECK FOR BEST SET OF INDEPENDENT COORDINATES
C         IFLAG = 2 WITHIN AN INTEGRATION LOOP DON'T CHANGE SET OF
C                     INDEPENDENT COORDINATES BEING USED
C         ANALOGOUS TO SWITCHING EULER ANGLE SEQUENCE WHEN GIMBAL
C         LOCK IS APPROACHED
C
C         PROCEED TO INVERT SUB-PARTITION OF BW ARRAY AND SOLVE FOR
C         DEPENDENT COORDINATES IF IFLAG=1 OR 2, INDEPENDENT
C         COORDINATES IF IFLAG=0
      DO 10 L=1,NCN
      IF(ICON(L).EQ.0) GO TO 10
      IST = 0
      LL = 1 + (L-1)/6
      LS = 6*(LL-1) + 3
      JST = JBIGST(LL)
      IF(L.GT.LS) JST = JST+3
      JEN = JBIGFN(LL)
  400 JBIG = 0
      A = ZRO
      DO 15 J=JST,JEN
      IF (JCON(J) .EQ. 0) GO TO 15
      AT = DABS(BW(L,J))
      IF (AT .LT. A) GO TO 15
      A = AT
      JBIG = J
   15 CONTINUE
      IVEC(L) = JBIG
      IF(A .GT. EPS) GO TO 400
      IF(A .LE. EPS.AND.IST.EQ.1) GO TO 999
      IST = 1
      JST = 1
      JEN = NU
      GO TO 400
  405 F = BW(L,JBIG)
      DO 17 J=1,NU
   17 BW(L,J) = BW(L,J)/F
      R(L) = R(L)/F
      BW(L,JBIG) = ONE
      DO 25 I=1,NCN
      IF (I .EQ. L   .OR. ICON(I) .EQ. 0) GO TO 25
      F = BW(I,JBIG)
      DO 30 J=1,NU
   30 BW(I,J) = BW(I,J) - F*BW(L,J)
      R(I) = R(I) - F*R(L)
      BW(I,JBIG) = ZRO
   25 CONTINUE
   10 CONTINUE
C
C          LEAD COORDINATES COMPUTED IN Y ARRAY
      DO 35 L=1,NCN
      LU = IVEC(L)
      IF (LU .EQ. 0) GO TO 35
      Y(LU) = ZRO
   35 CONTINUE
C
      DO 40 L=1,NCN
      IF (ICON(L) .EQ. 0) GO TO 40
      DO 45 J=1,NU
   45 R(L) = R(L) - BW(L,J)*Y(J)
   40 CONTINUE
      DO 50 L=1,NCN
      LU = IVEC(L)
      IF (LU .EQ. 0) GO TO 50
C          SET
C                INDEP(LU) = 0
C          IF COORDINATE LU IN STATE VECTOR IS DEPENDENT
      IF (IFLAG .EQ. 1) INDEP(LU) = 0
      Y(LU) = R(L)
   50 CONTINUE
C
      IF(.NOT.LEQU) RETURN
      WRITE (NOT,1004)
 1004 FORMAT (' INDEP ARRAY IS ')
      CALL WRITIS(INDEP,1,NU,1)
      RETURN
  999 WRITE (NOT,1001)
 1001 FORMAT (1H1,25HSINGULAR EQUATIONS, FINDU)
      STOP
C
      END




      SUBROUTINE YOUT
C                                                        DEBUG #   39
```

```
      IMPLICIT REAL*8(A-H,O-Z)                                              0 3742
C                                                                           0 3743
          COMMON /AMODA /                                                   0 3744
     *    ARO(2,2 , 6),BA(30,113)                                           1 3745
          COMMON /BBOSRD/                                                   0 3746
     *    BB(6,15,11),BS(6,15,15),BDE(3,3, 6),BBE(3, 6)                     2 3747
          COMMON /BANDS /                                                   0 3748
     *    BATH(3,12,10),SIGH(3,12,12),BATS(3,12,15),SIGS(3,12,15)           4 3749
          COMMON /ICINCR/                                                   0 3750
     *    ICINCR                                                            0 3751
          COMMON /INTGRL/                                                   0 3752
     *    AK(171, 6),ACCF(9,12, 6),RCOF(6,12, 6),                           5 3753
     *    COF11(12,12, 6),COF22(12,12, 6),COF33(12,12, 6),AK(12,12, 6),     6 3754
     *    COF12(12,12, 6),COF13(12,12, 6),COF23(12,12, 6),AU(12,12, 6),     7 3755
     *    COFAY(12,12, 6),COFAZ(12,12, 6),COFYZ(12,12, 6)                   8 3756
          COMMON /IVCONS/                                                   0 3757
     *    IV(6, 6)                                                          9 3758
          COMMON /JICPLS/                                                   0 3759
     *    JIC                                                               0 3760
          COMMON /LAMBDA/                                                   0 3761
     *    ALAM(30)                                                         10 3762
          COMMON /MAXMOD/                                                   0 3763
     *    NHMAX,NHMAX,NBPMAX,N4MAX,NMROD,NMODES,N4S,KY,KU                   0 3764
          COMMON /MEMENG/                                                   0 3765
     *    P(113),PMOM(30),HTOT(3),TOTE(3),ENGKE( 6),ENGPE( 6),             11 3766
     *    ETAKE, TOTPE, TOTENG, AMTOT,ATOTE                                 0 3767
          COMMON /NOPBRS/                                                   0 3768
     *    ZRC,ZNC,TAC,TRES                                                  0 3769
          COMMON /SPECIF/                                                   0 3770
     *    BETAH(6, 6),BETAHO(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),        16 3771
     *    BH(3,30),BS(3,30),IMO(3, 5),NYCA(6, 6),IFISHA(15),               17 3772
     *    NS,NH,NOPT,NOFMO,NDELTA,ITOPOL(2, 6),ENGFLA( 6),IHDATA(7, 6),    18 3773
     *    ESCC(14),EEND(14),NO,NDELTA,NLAM,NEQ                             19 3774
          COMMON /TAPENO/                                                   0 3775
     *    NTAPE1,NTAPE2,NTAPE3                                              0 3776
          COMMON /TIMESS/                                                   0 3777
     *    START,DELTAT,T, NOT,TAST                                          0 3778
          COMMON /VECTOR/                                                   0 3779
     *    Y(250),YDT(250)                                                  20 3780
C                                                                           0 3781
      DIMENSION BSV(113),D(30),V(30),HDTQ(6,18),BBTP(6,18)                 75 3782
C                                                                           0 3783
      DIMENSION BMB(36,36),BM(6,22,11)                                     76 3784
      EQUIVALENCE (BW(1),BMB(1)),  (BW(1297),BM(1))                        77 3785
      EQUIVALENCE (ALAM(1),V(1))                                            0 3786
C                                                                           0 3787
      DATA IFLAG / 1 /                                                      0 3788
      DATA NOT/6 /                                                          0 3789
C                                                                           0 3790
      LOGICAL DEBUG,LEQU                                                    0 3791
      COMMON /LDEBUG/ DEBUG(120)                                            0 3792
      EQUIVALENCE (LEQU,DEBUG(39))                                          0 3793
      IF(LEQU) WRITE(NOT,1000)                                              0 3794
 1000 FORMAT (' SUBROUTINE YOUT')                                           0 3795
      IF (JIC .EQ. 4) IFLAG = 1                                             0 3796
C                                                                           0 3797
      IF(.NOT.LEQU) GO TO 21                                                0 3798
      WRITE (NOT,1002)                                                      0 3799
 1002 FORMAT (' THE STATE VECTOR Y IS ')                                    0 3800
      CALL WRITES(Y,1,NEQ,1)                                                0 3801
```

```
   21 CONTINUE
C
C         GET DATA TO DEFINE HINGE AND BODY FIXED FRAME KINEMATICS
      CALL ROTSH
C
C         GENERATE HH ARRAY
      CALL HHGENR
C
C         GET DATA TO DEFINE SENSOR POINT AXIS SYSTEM ORIENTATION
      CALL ROTCS
C
C         GET DATA TO DEFINE KINEMATICAL COEFFICIENTS FOR SENSOR POINTS
      CALL SSGENR
C
C         BRANCH TO MGEN TO GENERATE DEFORMATION DEPENDENT MASS MATRIX
C         AND GRAVITY GRADIENT DATA, SEE VOL I PAGES 45-52
      CALL MGEN
C
C         IF NLAM(NUMBER OF CONSTRAINTS) GREATER THAN ZERO BRANCH TO
C         FINDU(I) TO GET DEPENDENT COORDINATES TO PUT INTO STATE
C         VECTOR ARRAY Y
C         IFLAG = 2 ON INTERMEDIATE STEP OF RUNGE KUTTA CYCLE
      IF (NLAM .GT. 0) CALL FINDU (IFLAG)
C
      DO 10 L=1,NB
      LU = LOCU(L)
      LE = LENU(L)
C
C         GET ORDINARY MOMENTA EQUATION II-49 VOL I
   10 CALL MULTS  (AMU(1,1,L),Y(LU),P(LU),LE,LE,1,KMU,1,1)
      DO 11 I=1,NU
   11 GGV(I) = ZRO
C
C         GET GRAVITY GRADIENT DATA
      CALL GRVGRD (GGV)
      DO 15 L=1,NB
      LE = LENU(L)
C
C         INVERT THE MASS MATRIX FOR BODY L STORE RESULT IN AMU ARRAY
C         MASS MATRIX DESTROYED HERE
   15 CALL INVINP (AMU(1,1,L),AMU(1,1,L),LE,KMU)
      IF (NLAM .EQ. 0) GO TO 200
C
C         GET B*M**(-1)*B**T FOR USE IN OBTAINING LAGRANGE MULTIPLIERS
      CALL GETEMB
      KBMB = 6*NHMAX
C         DECOMPOSE BMB MATRIX
      CALL DECM2  (BMB,D,NLAM,KBMB)
C
C
CC   CALCULATE BETADT AND PLACE INTO YDT
C         COMPUTE BETA DOT TERM OF EQUATION II-3 VOL I PUT IN YDT ARRAY
C         ALSO SAVE IT IN BETAHD ARRAY
  200 IC = LOCU(2*NH+1) - 1
      DO 60 L=1,NH
      DO 60 I=1,6
      IP1 = I + 1
      IF (IFDATA(IP1,L) .EQ. 1) GO TO 60
      IC = IC + 1
      YDT(IC) = ZRO
      IF (L .EQ. 1) GO TO 61
      NOBQ = ITOPOL(1,L)
      NOBP = ITOPOL(2,L)
      LQ = 2*L - 2
      LP = LQ + 1
      LOQ = LOCO(NOBQ) - 1
      LOP = LOCO(NOBP) - 1
      LEQ = IRGFLX(NOBQ) + 6
      LEP = IRGFLX(NOBP) + 6
C
C        EVALUATE ALL BETA DOT TERMS IN EQUATION II-3 STACK THEM
C        IN YDT ARRAY AFTER MODAL VELOCITIES
      DO 62 J=1,LEQ
      LOQJ = LOQ + J
   62 YDT(IC) = YDT(IC) + BH(I,J,LQ)*Y(LOQJ)
      DO 63 J=1,LEP
      LOPJ = LOP + J
   63 YDT(IC) = YDT(IC) + BH(I,J,LP)*Y(LOPJ)
      BETAFD(I,L) = YDT(IC)
      GO TO 60
   61 LEQ = IRGFLX(1) + 6
      DO 64 J=1,LEQ
   64 YDT(IC) = YDT(IC) + BH(I,J,1)*Y(J)
      BETAFD(I,L) = YDT(IC)
   60 CONTINUE
C
C  PUT MODAL VELOCITIES (XI(DOT)) INTO YDOT
C        THESE GO AFTER BODY ACCELERATIONS. THEY CAN BE FOUND IN THE
C        STATE VECTOR SEE ARRANGEMENT OF U VECTOR
C
      DO 65 N=1,NB
      LL = IRGFLX(N)
      IF (LL .EQ. 0) GO TO 65
      LOJ = LOCO(N) + 5
      LO = LOCO(N+NB) - 1
      DO 66 J=1,LL
   66 YDT(LO+J) = Y(LOJ+J)
   65 CONTINUE
C
C        BRANCH TO TORQUE TO COMPUTE ALL TIME DEPENDENT FORCES FOR ALL
C        BODIES.GGV ARRAY CONTAINS SUM OF ALL FORCES.SEE VOL 1,PP6-8
      CALL TORQUE (GGV)
C
CC  INITIALIZE GDOT. GET RHS OF LAMBDA EQUATION
C
C        COMPUTE M**(-1)*G PUT RESULT IN APPROPRIATE ROW OF YDT
      DO 70 N=1,NB
      LF = LOCO(N)
      LL = LENG(N)
   70 CALL MULTS  (AMO(1,1,N),GGV(LF),YDT(LF),LL,LL,1,KMU,1,1)
      IF(.NOT.LLGO) GO TO 20
      WRITE(NOT,1001)
 1001 FORMAT (' YDT ARRAY SANS CONSTRAINT TORQUES  ')
      CALL WRITES(YDT,1,NEQ,1)
   20 CONTINUE
C
      IFLAG = 2
      IF (NCAM .EQ. 0) RETURN
C
C        IF FIXED OR RHEONOMIC CONSTRAINTS EXIST EVALUATE ALL LAGRANGE
```

```
C        MULTIPLIERS VIA EQUATION II-6 VOL 1
      DO 80 L=1,NH
C        BRANCH TO BDOTQP TO GET BDTQ AND BDTP, THE LOWER CASE B DOT
C        MATRIX IN EQUATION II-6 VOL 1
      CALL BDOTQP (L,BDTQ,BDTP)
      DO 80 I=1,6
      IP1 = I + 1
      IC = IV(I,L)
      IF (IC .EQ. 0) GO TO 80
      V(IC) = ZRO
      IIC = 6*(L-1) + I
C        IF RHENOMIC CONSTRAINTS CALL FUNCTION ADDT TO DEFINE TIME
C        DEPENDENT ACCELERATION, THE ALPHA DOUBLE DOT TERM IN EQ. II-6
      IF (IFDATA(IP1,L) .EQ. 2) V(IC) = ADDT(IIC,1)
      IF (L .EQ. 1) GO TO 81
      NOBQ = ITOPOL(1,L)
      NOBP = ITOPOL(2,L)
      LQ = 2*L - 2
      LP = LQ + 1
      LOQ = LOCQ(NOBQ) - 1
      LOP = LOCQ(NOBP) - 1
      LEQ = IRGFLX(NOBQ) + 6
      LEP = IRGFLX(NOBP) + 6
C        GET REST OF TERMS WITHIN PARENTHESES IN EQUATION II-6
      DO 82 J=1,LEQ
      LOQJ = LOQ + J
   82 V(IC) = V(IC) - BH(I,J,LQ)*YDT(LOQJ) - BDTQ(I,J)*Y(LOQJ)
      DO 83 J=1,LEP
      LOPJ = LOP + J
   83 V(IC) = V(IC) - BH(I,J,LP)*YDT(LOPJ) - BDTP(I,J)*Y(LOPJ)
      GO TO 80
   81 LEQ = IRGFLX(1) + 6
      DO 84 J=1,LEQ
   84 V(IC) = V(IC) - BH(I,J,1)*YDT(J) - BDTQ(I,J)*Y(J)
   80 CONTINUE
C
      IF (NLAM .GT. 1) GO TO 305
      V(1) = V(1)/D(1)
      GO TO 310
C        USE DECOMPOSED GMB AND D ALONG WITH V IN DAKSEV TO SOLVE II-6
C        AND RETURN LAGRANGE MULTIPLIERS IN V ARRAY
  305 CALL DAKSEV (GMB,NLAM,V,D,KGMB)
C
  310 LEQ = LENQ(1)
C        FILL OUT EQUATION II-1 ACCOUNTING FOR ACTION OF CONSTRAINT
C        FORCES. THE GM ARRAY WAS OBTAINED IN GETGMB
      DO 85 I=1,6
      ICN = IV(I,1)
      IF (ICN .EQ. 0) GO TO 85
      F = V(ICN)
      DO 86 J=1,LEQ
   86 YDT(J) = YDT(J) + F*GM(I,J,1)
   85 CONTINUE
C
      DO 90 L=2,NH
      NOBQ = ITOPOL(1,L)
      NOBP = ITOPOL(2,L)
      LQ = 2*L - 2
      LP = LQ + 1
      LOQ = LOCQ(NOBQ) - 1
      LOP = LOCJ(NEQP) - 1                                                 0 3982
      LEQ = LENG(NEQU)                                                     0 3983
      LEP = LENG(NEQP)                                                     0 3984
      DO 90 I=1,6                                                          0 3985
      IEN = IV(I,L)                                                        0 3986
      IF (IEN .EQ. 0) GO TO 90                                             0 3987
      F = Y(IEN)                                                           0 3988
      DO 80 J=1,LEQ                                                        0 3989
      LOQJ = LOQ + J                                                       0 3990
   80 YDT(LOQJ) = YDT(LOQJ) + F*EQM(I,J,LQ)                                0 3991
      DO 85 J=1,LEP                                                        0 3992
      LOPJ = LOP + J                                                       0 3993
   85 YDT(LOPJ) = YDT(LOPJ) + F*EPM(I,J,LP)                                0 3994
   90 CONTINUE                                                             0 3995
C                                                                          0 3996
      IF(.NOT.LDBUG) RETURN                                                0 3997
      WRITE(NOT,1003)                                                      0 3998
 1003 FORMAT (' THE YDT ARRAY IS ')                                        0 3999
      CALL WRITES (YDT,1,NEQ,1)                                            0 4000
      RETURN                                                               0 4001
      END                                                                  0 4002




      SUBROUTINE LINEAR (NR,NC)                                            0 4003
C                                                          DEBUG *  40     0 4004
      IMPLICIT REAL*8 (A-H,O-Z)                                            0 4005
C                                                                          0 4006
C  SUBROUTINE ESTABLISHES FIRST PARTIAL DERIVATIVES OF A YDOT              0 4007
C  DESCRIBED FUNCTIONAL AT AN INITIAL STATE, Y, USING QUADRATIC            0 4008
C  INTERPOLATION FUNCTIONS.                                                0 4009
C          REFER TO VOL I SECTION III 'SYNTHESIS AND ANALYSIS OF THE       0 4010
C            LINEARIZED SYSTEM' FOR THEORETICAL DEVELOPMENT                0 4011
C                                                                          0 4012
C  THE PARTIAL DERIVATIVE MATRIX IS WRITTEN COLUMNWISE ON UNIT NO.         0 4013
C  CALLS SUBROUTINE YDOT                                                   0 4014
            COMMON /PRADRS/                                                0 4015
     *      PR(250,5)                                                     14 4016
            COMMON /TAPENO/                                                0 4017
     *      NTAPE1,NTAPE2,NTAPE3                                           0 4018
            COMMON /VECTOR/                                                0 4019
     *      Y(250),YDT(250)                                               20 4020
            COMMON /VINDEP/                                                0 4021
     *      INDEP(250)                                                    21 4022
      COMMON /EQSIZE/ NX,NY,DELTA,NESS,NT,NJC,NEQ,NJE                      0 4023
C                                                                          0 4024
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                  0 4025
C  NR       = INPUT  NUMBER OF ROWS IN PARTIAL DERIVATIVE MATRIX.          0 4026
C  NC       = INPUT  NUMBER OF COLUMNS IN PARTIAL DERIVATIVE MATRIX.       0 4027
C       (FROM DYNS2)                                                       0 4028
C          NC = NEQ                                                        0 4029
C          NR = NEQ + NAUX                                                 0 4030
C                                                                          0 4031
      DIMENSION FY(250,3),Z(250),ZNEW(250),IV(250)                        78 4032
      EQUIVALENCE (PR(1),FY(1)),(PR( 751),Z(1)),(PR(1001),ZNEW(1))        79 4033
      EQUIVALENCE (INDEP(1),IV(1))                                         0 4034
      DATA NOT/ 6 /                                                        0 4035
      DATA PCEN,PMIN/ 1.D-02,1.D-04 /
C
C******************** CAUTION EPS1 CAN BE PROBLEM SENSITIVE
C
      DATA EPS1,EPS2/ 1.D-06,1.D-04 /
      COMMON /LDEBUG/ LDEBUG(120)
      LOGICAL LDEBUG,LEQU
      EQUIVALENCE (LEQU,LDEBUG(43))
C
      IF(LEQU) WRITE(NOT,1000)
 1000 FORMAT (' SUBROUTINE LINEAR ')
      NC = NTAPE2
C
C   ESTABLISH OUTPUT SIZE OF PARTIAL DERIVATIVE MATRIX
C
      NJG = 0
      NA  = 0
C
      DO 5 I=1,NR
C           IV(I)=0 IF COORDINATE I IN STATE VECTOR Y A DEPENDENT COORD.
      IF (IV(I) .NE. 0) NJG = NJG+1
      IF (I .GT. NC) GO TO 5
      IF (IV(I) .NE. 0) NA=NA+1
    5 CONTINUE
C
      REWIND NC
C
C           LOAD FY(1) WITH YDT, ETA=0
      DO 20 I=1,NR
   20 FY(I,1) = YDT(I)
C
      DO 200 L=1,NC
      IF (IV(L) .EQ. 0) GO TO 200
      IFLG = 0
      EPSV = EPS1
   32 CONTINUE
      DY = PCEN * Y(L)
      IF(DABS(DY).LT.PMIN) DY=PMIN
C
      YY = Y(L)
      Y(L) = Y(L) + DY
      CALL YDOT
      Y(L) = YY
C
C           LOAD FY(3) WITH YDT, ETA=1
      DO 30 I=1,NR
   30 FY(I,3) = YDT(I)
C
      YY = Y(L)
      Y(L) = Y(L) + 0.5D0 * DY
      CALL YDOT
      Y(L) = YY
C
C           LOAD FY(2) WITH YDT, ETA=.5
      DO 35 I=1,NR
   35 FY(I,2) = YDT(I)
C
      ZERO = 0.0D0
      DO 50 I=1,NR
C           EVALUATE EQUATION III-10
```

```
      E1 = -3.D0 * FY(I,1) + 4.D0 *FY(I,2) - FY(I,3)
C        EVALUATE EQUATION III-17
C            ALSO LOOK FOR LARGEST Z(I)
      Z(I) = E1 / DY
      ZAB = DABS(Z(I))
      IF(ZAB.LT.EPS1) Z(I) = 0.0D0
      IF(ZAB.GT.ZERO) ZERO = ZAB
   50 CONTINUE
      ITR = 0
C
   60 DY = 0.5D0*DY
C        INCREMENT SIZE HAS BEEN CUT IN HALF
C
C        LOAD F(3) WITH YDT, ETA=.5
      DO 70 I=1,NR
   70 FY(I,3) = FY(I,2)
C
      YY = Y(L)
      Y(L) = Y(L) + 0.5D0 * DY
      CALL YDOT
      Y(L) = YY
C
C        LOAD FY(2) WITH YDT, ETA=.25
      DO 80 I=1,NR
   80 FY(I,2) = YDT(I)
C
      DO 90 I=1,NR
C        EVALUATE EQUATION III-16
      E1 = -3.D0 * FY(I,1) + 4.D0 * FY(I,2) - FY(I,3)
C        EVALUATE EQUATION III-17
      ZNEW(I) = E1/DY
      IF (DABS(ZNEW(I)) .LT.  EPS1)   ZNEW(I) = 0.D0
   90 CONTINUE
C
C        COMPARE Z AND ZNEW TO SEE IF NEARLY EQUAL
      IF(.NOT.LOGO) GO TO 21
      WRITE (NOUT,1002) ITR,L
      CALL WRITE3(Z,1,NR,1)
      WRITE (NOUT,1003) ITR,L
      CALL WRITE3 (ZNEW,1,NR,1)
 1002 FORMAT (' Z    ARRAY ITERATION',I3,' FOR COLUMN',I3)
 1003 FORMAT (' ZNEW ARRAY ITERATION',I3,' FOR COLUMN',I3)
   21 CONTINUE
      DO 100 I=1,NR
C
      IF (IV(I) .EQ. 0) GO TO 100
C
C        CONVERGENCE TEST
      DN = DABS(Z(I))
      DN1 = DABS(ZNEW(I))
C
      IF (DN1 .GT. DN) DN = DN1
      IF (DN  .LT. EPS1) GO TO 100
      C1 = DABS(ZNEW(I) - Z(I))/ DN
C
      IF (C1 .LE. EPS2) GO TO 100
C
      ITR = ITR + 1
C         GIVE UP AFTER 5 TRYS AND 2 EPS1 CHANGES
      IF(ITR.LE.5) GO TO 60
      IF(IFLG.GT.0) GO TO 31
      IFLG = 1
      EPSI = 1.0D-06*ZERO
      WRITE(NOT,2002) L,EPSI
      GO TO 32
   31 IF(IFLG.GT.1) GO TO 33
      IFLG = 2
      EPSI = 1.0D-04*ZERO
      WRITE(NOT,2002) L,EPSI
      GO TO 32
   33 EPSI = EPSV
      GO TO 999
   90 CONTINUE
      DO 95 J=1,NR
   95 Z(J) = ZNEW(J)
      GO TO 60
  100 CONTINUE
C
C  COMPLETION OF THE DO 100 LOOP INDICATES WE HAVE ACCEPTED
C  ZNEW(I) ,I=1,NR.
C
C  NOW PACK PARTIAL DERIVATIVES INTO A NJG LONG VECTOR.
C
      J=0
      DO 110 I=1,NR
      IF (IV(I) .EQ.0) GO TO 110
      J=J+1
      ZNEW(J) = ZNEW(I)
  110 CONTINUE
C
      WRITE (NC) (ZNEW(J),J=1,NJG)
C
      IF(.NOT.ECHO) GO TO 220
      WRITE(NOT,1001) L
 1001 FORMAT (' COLUMN',I3,' OF THE PARTIAL DERIVATIVE MATRIX IS')
      CALL WRITES(ZNEW,1,NJG,1)
C
  220 CONTINUE
      RETURN
C
  999 WRITE (NOT,2001) I,L,DY,Z(I),ZNEW(I)
 2001 FORMAT (1H1,////,20X,
     * 30HSUBROUTINE LINEAR FAILED TO CONVERGE ,/,20X,
     * 29HIN  5 ITERATIONS ON ELEMENT ,/,10X,
     *4HI = ,I5,/,10X,
     *4HL = ,I3,/,10X,19HLAST Y INCREMENT = ,D12.4,//,10X,
     *7HZ     = D12.4,/,10X,
     *7HZNEW = D12.4)
 2002 FORMAT (10X,'CONVERGENCE PROBLEMS IN COLUMN',I3,' OF LINEAR EPSI R
     +ESET TO',D12.4,' FOR THIS COLUMN')
C
      STOP
      END
      SUBROUTINE INVINP (A,R,N,KA)                                      C 4209

C                                                  DEBUG *   41         C 4210
C        ROUTINE TO INVERT THE MASS MATRIX                              C 4211
      IMPLICIT REAL*8(A-H,O-Z)                                          C 4212
      LOGICAL [illegible]                                               C 4213
      COMMON [illegible]                                                C 4214
      DIMENSION [illegible] (KA,1),C(41))                               C 4215
C                                                                       C 4216
C*********                                                              C 4217
C              MATRIX A MUST BE SYMMETRIC, AND N = KA                   C 4218
C*********                                                              C 4219
C                                                                       C 4220
      EQUIVALENCE [illegible](R(1,1)),(R(KA,1)),C( ))                  HC 4221
C                                                                       C 4222
      DATA  EPS,NOT / 1.D-30, 6/                                        C 4223
C                                                                       C 4224
      IF(.NOT.LLDU) GO TO 1                                             C 4225
      WRITE(NOT,1000)                                                   C 4226
      WRITE(NOT,1001)                                                   C 4227
      CALL WRITES(A,N,N,KA)                                             C 4228
    1 CONTINUE                                                          C 4229
      L = 1                                                             C 4230
      IF (A(1,1) .LT. EPS) GO TO 999                                    C 4231
C                                                                       C 4232
      R(1,1) = 1.D+00/A(1,1)                                            C 4233
      DO 100 L = 2,N                                                    C 4234
      M1 = L - 1                                                        C 4235
      DO 10 I=1,M1                                                      C 4236
      C(I) = 0.D+00                                                     C 4237
      DO 10 J=1,M1                                                      C 4238
   10 C(I) = C(I) + R(I,J)*A(J,L)                                       C 4239
      Z = A(L,L)                                                        C 4240
      DO 20 I=1,M1                                                      C 4241
   20 Z = Z - A(I,L)*C(I)                                               C 4242
      IF (DABS(Z) .LT. EPS) GO TO 999                                   C 4243
      Z = 1.D+00/Z                                                      C 4244
      DO 30 I=1,M1                                                      C 4245
      V = -Z*C(I)                                                       C 4246
      R(I,L) = V                                                        C 4247
      R(L,I) = R(I,L)                                                   C 4248
      DO 30 J=1,M1                                                      C 4249
      R(I,J) = R(I,J) - V*C(J)                                          C 4250
   30 R(J,I) = R(I,J)                                                   C 4251
      R(L,L) = Z                                                        C 4252
  100 CONTINUE                                                          C 4253
      IF(.NOT.LLDU) RETURN                                              C 4254
      WRITE(NOT,1002)                                                   C 4255
      CALL WRITES(R,N,N,KA)                                             C 4256
 1000 FORMAT (' SUBROUTINE INVINP ')                                    C 4257
 1001 FORMAT (' MASS MATRIX IS ')                                       C 4258
 1002 FORMAT (' INVERSE OF MASS MATRIX IS ')                            C 4259
C                                                                       C 4260
      RETURN                                                            C 4261
C                                                                       C 4262
  999 WRITE(NOT,903) L                                                  C 4263
  903 FORMAT(/,5X,32HSINGULAR MATRIX IN INVINP AT L =,I3,9H STOP RUN)   C 4264
      STOP                                                              C 4265
C                                                                       C 4266
      END                                                               C 4267
      SUBROUTINE DECM2 (U,D,N,NR)
C                                                         DEBUG #  42
      IMPLICIT REAL*8(A-H,O-Z)
C        DECOMPOSE U INPUT INTO UPPER TRIANGULAR ELEMENTS OF U AND
C        ELEMENTS OF THE DIAGONAL MATRIX D SUCH THAT
C              U(INPUT) = U**T*D*U
C              DIAGONAL AND LOWER TRIANGULAR ELEMENTS OF OUTPUT U
C              CONTAIN GARBAGE
C              U SHOULD HAVE ONES ON DIAGONAL AND ZEROS BELOW IT
C              NOT DONE HERE TO SAVE TIME, TAKSEV KNOWS THIS
C        ALGORITHM USED ASSUMES THAT U IS SYMMETRIC
      DIMENSION U(NR,1),D(1)
      DATA EPS,NOT /1.D-15, 6/
C
      IF (N .EQ. 1) GO TO 20
      NM1 = N - 1
      DO 15 L=1,NM1
      IF (DABS(U(L,L)) .LT. EPS) GO TO 998
      LP1 = L + 1
      D(L) = U(L,L)
      DO 15 I=LP1,N
      S = U(L,I)/D(L)
      DO 10 J=I,N
   10 U(I,J) = U(I,J) - S*U(L,J)
   15 U(L,I) = S
   20 D(N) = U(N,N)
      RETURN
C
  998 WRITE (NOT,1001)
 1001 FORMAT (1H1,40HMATRIX SINGULAR, DECM2, PROGRAM STOPPED.)
      STOP
      END
```

```
      SUBROUTINE WRITES (A,NR,NC,KR)
C                                                         DEBUG #  43
C      PRINT REAL TWO DIMENSIONAL ARRAY IN FORMA FORMAT
      REAL*8 A
      DIMENSION A(KR,1),ICHEAD(10)
      DATA NOT / 6/
      DATA ICHEAD/4H(  1),4H(  2),4H(  3),4H(  4),4H(  5),
     *             4H(  6),4H(  7),4H(  8),4H(  9),4H( 10)  /
C
 2010 FORMAT ( 8X,10(7X,A4))
 2030 FORMAT (1X,2I5,2X,1P10D11.3)
C
      LR = 10
      IF (NC .LT. LR) LR = NC
      WRITE (NOT,2010)  (ICHEAD(L),L=1,LR)
      DO 60 I=1,NR
      JS = 1
   10 JE = JS + 9
      IF (JE .GT. NC) JE = NC
      WRITE (NOT,2030) I,JS, (A(I,J),J=JS,JE)
      IF (JE .EQ. NC) GO TO 60
      JS = JS + 10
      GO TO 10                                                          0 4322
   60 CONTINUE                                                          0 4323
C                                                                       0 4324
      RETURN                                                            0 4325
      END                                                               0 4326




      SUBROUTINE ROTOR                                                  0 4327
C                                                     DEBUG #  44       0 4328
      IMPLICIT REAL*8(A-H,O-Z)                                          0 4329
C                                                                       0 4330
C        USED TO COMPUTE ARRAYS REL AND COL SEE VOL II,PAGE 18          0 4331
C        INTERMEDIATE ROTATIONS STORED IN RH                            0 4332
C        VELOCITY TRANSFORMATION STORED IN PIN                          0 4333
C                                                                       0 4334
              COMMON /BRUSH3/                                           0 4335
     *      DH(6,12,11),DS(6,18,15),REL(3,3, 6),COL(3, 6)               2 4336
              COMMON /HANDS /                                           0 4337
     *      RATE(3,12,15),SIGM(3,12,15),RATS(3,12,15),SIGS(3,12,15)     4 4338
              COMMON /PINRP /                                           0 4339
     *      PIN(3,3, 6), RP2(3,3, 6), RP3(3,3, 6)                      13 4340
              COMMON /SPECIF/                                           0 4341
     *      DETAM(6, 6),DETAMJ(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),  16 4342
     *      DH(3,30),DS(3,30),IMO(3, 5),NMCA(6, 3),[illegible](15),     17 4343
     *      NT,NH,NSPT,NORMC,NDELTA,ITCHDL(2, 5),IROFLAG( 6),IMDATA(7, 6), 18 4344
     *      ESCC(14),ECNG(14),ND,NDELTA,NEAR,NEQ                       19 4345
              COMMON /VECTOR/                                           0 4346
     *      V(250),VDT(250)                                            20 4347
C                                                                       0 4348
      DIMENSION DF(3),ANGF(3),ROT(3,3),RM(3,3),                         0 4349
     *         ITRM(2, 6),IORT(2, 6),NSDF( 5)                          81 4350
C                                                                       0 4351
      DATA IIST,NCT / 0, 0/                                             0 4352
      LOGICAL [illegible]                                               0 4353
      COMMON /ZDEBUG/ DEBUG(120)                                        0 4354
      EQUIVALENCE (ID,DEBUG(44))                                        0 4355
C                                                                       0 4356
      IF (DEBUG) WRITE (6,1000)                                         0 4357
 1000 FORMAT (' SUBROUTINE ROTOR ')                                     0 4358
      IF (IIST .EQ. 1) GO TO 500                                        0 4359
      IIST = 1                                                          0 4360
      DO 100 I=1,2                                                      0 4361
      DO 100 J=1,NH                                                     0 4362
  100 ITRM(I,J) = ITCHDL(I,J)                                           0 4363
      [illegible]                                                       0 4364
      DO 120 [illegible]                                                0 4365
  120 [illegible]                                                       0 4366
      [illegible]                                                       0 4367
      [illegible]                                                       0 4368
      IORT(1,1) = 1                                                     0 4369
      IORT(2,1) = 1                                                     0 4370
      DO 115 I=1,2                                                      0 4371
      DO 115 J=1,NH                                                     0 4372
      IF (ITRM(I,J) [illegible]) GO TO 115                              0 4373
      [illegible]                                                       0 4374
      [illegible]                                                       0 4375
      IOP(2,J) = ITOP*(1,J)
      IF (I .EQ. 2) GO TO 115
      IOP(1,J) = -NOP
      IOP(2,J) = ITOP*(2,J)
  115 IS1 = IOP(2,J)
      NSCOF(IS1) = 1
      ITOP*(1,J) = 0
      ITOP*(2,J) = 0
  110 CONTINUE
      NSCOF(IS) = 0
      IF (NOP .EQ. NM) GO TO 120
      DO 116 N=1,NB
      IF (NSCOF(N) .EQ. 0) GO TO 116
      IS = N
      GO TO 125
  116 CONTINUE
      GO TO 955
  120 DO 130 J=1,NM
      JOPA = IOP(1,J)
      JOP = IABS(JOPA)
      ITOP*(1,JOP) = J*JOPA/JOP
  130 ITOP*(2,JOP) = IOP(2,J)
      IF(.NOT.LECHO) GO TO 80
      WRITE (NCT,1331)
 1331 FORMAT (' ITOP* ARRAY IS ')
      CALL WRITIS(ITOP*,2,NM,2)
   80 CONTINUE
C
  500 CONTINUE
C          LOAD ELTAP WITH UPDATED ANGLES FROM Y STATE VECTOR
      NDLT = LLCO(2*NB + 1) - 1
      DO 5 J=1,NM
      DO 5 I=1,6
      II = I + 1
      IF (IFDATA(II,J) .EQ. 1) GO TO 5
      NDLT = NDLT + 1
      ELTAP(I,J) = Y(NDLT)
    5 CONTINUE
C
      DO 10 I=2,NM
      NOBG = ITOPOL(1,I)
      NOBP = ITOPOL(2,I)
C          LR1 = 1,7,13,19,....
C          LC1 = 1,8,15,22,....
      LR1 = 1 + 6*(I-2)
      LC1 = 1 + 7*(I-2)
      LR2 = LR1 + 1
      LR3 = LR1 + 2
      LR4 = LR1 + 3
      LR5 = LR1 + 4
      LR6 = LR1 + 5
      LC2 = LC1 + 1
      LC3 = LC1 + 2
      LC4 = LC1 + 3
      LC5 = LC1 + 4
      LC6 = LC1 + 5
      LC7 = LC1 + 6
      NMG = INDEX(NOBG)
      NMP = INDEX(NOBP)
C
```

```
C     GO TO 15 IF BODY CONTAINING Q FRAME RIGID
      IF (NMQ .EQ. 0) GO TO 15
C
C     ACCOUNT FOR BODY FLEXIBILITY TO GET POSITION DM(LC3) AND
C          THE ROTATION RM(LR3) OF Q FRAME AT HINGE I WRT ITS BODY FRAME
      [illegible]
      [illegible] - 3
      CALL MULTS (PATM(1,1,LMQ),Y(LQ),DF,3,NMQ,1,3,1,1)
      CALL MULTS (SIGM(1,1,LMQ),Y(LQ),ANGF,3,NMQ,1,3,1,1)
      CALL [illegible] (3,1,ANGF,ANGTF,LDUM,LDUM)
      CALL MULTS (ROTF,RM(1,1,LR1),RM(1,1,LR3),3,3,3,3,3,3)
      DO 12 J=1,3
   12 DM(J,LC3) = DM(J,LC1) + DF(J)
C
C     GO TO 20 IF BODY CONTAINING P FRAME RIGID
   15 IF (NMP .EQ. 0) GO TO 20
C
C     ACCOUNT FOR BODY FLEXIBILITY TO GET POSITION DM(LC4) AND
C          THE ROTATION RM(LR4) OF P FRAME AT HINGE I WRT ITS BODY FRAME
      [illegible]
      [illegible] - 3
      CALL MULTS (PATM(1,1,LMP),Y(LP),DF,3,NMP,1,3,1,1)
      CALL MULTS (SIGM(1,1,LMP),Y(LP),ANGF,3,NMP,1,3,1,1)
      CALL [illegible] (3,1,ANGF,ANGTF,LDUM,LDUM)
      CALL MULTS (ROTF,RM(1,1,LR2),RM(1,1,LR4),3,3,3,3,3,3)
      DO 17 J=1,3
   17 DM(J,LC4) = DM(J,LC2) + DF(J)
C
      [illegible]
      [illegible]
      [illegible] = [illegible]TX(J,I)
      [illegible] = [illegible]TAM(J,I)
      IT = [illegible]IAT(I,I)
C
C     [illegible] RELATIVE ROTATION BETWEEN P AND Q FRAMES AT HINGE I
C          [illegible] INSTITUTE EULER ANGLES
      CALL [illegible] (3,IT,ANGF,RM(1,1,LR5),LDUM,LDUM)
C
C     GET THE [illegible] MATRIX OF EQUATION II-19 VOL I
C              [illegible](I,I) = ROTATION TRANSFORMATION MATRIX RELATING BODY
C                        FIXED ANGULAR RATES TO EULER RATES BETWEEN P AND
C                        Q FRAME AT HINGE I(INVERSE[illegible])
      CALL [illegible] (1,IT,ANGF,PIN(1,1,I),PH2(1,1,I),PH3(1,1,I))
C
      DO 35 L=1,3
      DO 35 J=1,3
   35 FA(L,J) = RM(J,L,LR3)
C
C          RM(LR6) = RM(LR4)*RM(LR5)*(RM(LR3))**T
C                   TRANSFORMATION BETWEEN BODY FIXED FRAMES OF BODIES
C                   CONTIGUOUS AT HINGE I
      CALL MULTS (RM(1,1,LR4),RM(1,1,LR5),ROTF,3,3,3,3,3,3)
      CALL MULTS (ROTF,FA,RM(1,1,LR6),3,3,3,3,3,3)
C
C          DM(LC6) = VECTOR BETWEEN BODY FIXED REFERENCE FRAMES OF
C                   BODIES CONTIGUOUS AT HINGE I
      CALL MULTS (RM(1,1,LR4),DM(1,LC5),DF,3,3,1,3,3,1)
      CALL MULTS (RM(1,1,LR6),DM(1,LC3),ANGF,3,3,1,3,3,1)
      DO 40 J=1,3
   40 DM(J,LC6) = DM(J,LC4) + DF(J) - ANGF(J)
```

```
C
   10 CONTINUE
C        ALL RELATIVE POSITION AND ORIENTATION DATA OF P-G FRAME AT
C        EACH HINGE POINT COMPUTED AND STORED IN OH AND RH ARRAYS
C
C
C
C        PROCEED TO COMPUTE:
C              ROL(K) ROTATION TRANSFORMATION MATRIX BODY K FIXED AXIS
C                     TO INERTIAL AXIS
C              DOL(K) POSITION VECTOR FROM INERTIAL ORIGIN TO BODY
C                     K FIXED REFERENCE
      DO 45 J=1,3
      JP3 = J + 3
      DOL(J,1) = BETAM(JP3,1)
   45 ANG(J) = BETAM(J,1)
      IT = ITA(1,1)
      CALL ROTTR (3,IT,ANG,ROL(1,1,1),DUM,DUM)
      CALL ROTTR (1,IT,ANG,RIN(1,1,1),RP2(1,1,1),RP3(1,1,1))
C
      DO 50 J=2,NH
      NOH = ITOPW(1,J)
      LROJ = ITOPW(2,J)
      IF (NOH .LT. 0) GO TO 52
      LRE = 9*(NOH - 1)
      LRO = ITOPOL(2,NOH)
      CALL MULT3  (ROL(1,1,LRO),RH(1,1,LRE),ROL(1,1,LROJ),3,3,3,3,3,3)
      GO TO 50
   52 NOH = -NOH
      LRE = 9*(NOH - 1)
      LRO = ITOPOL(1,NOH)
      DO 53 I=1,3
      DO 53 L=1,3
   53 RA(I,L) = RH(L,I,LRE)
      CALL MULT3 (ROL(1,1,LRO),RA,ROL(1,1,LROJ),3,3,3,3,3,3)
   50 CONTINUE
C
      DO 60 J=2,NH
      LRO = ITOPOL(2,J)
      LD6 = 7*(J-1) - 1
      LD7 = LD6 + 1
   60 CALL MULT3  (ROL(1,1,LRO),OH(1,LD6),OH(1,LD7),3,3,1,3,3,3)
C
      DO 70 J=2,NH
      NOH  = ITOPW(1,J)
      LDOJ = ITOPW(2,J)
      IF (NOH .LT. 0) GO TO 72
      LD7 = 7*(NOH - 1)
      LDO = ITOPOL(2,NOH)
      DO 74 I=1,3
   74 DOL(I,LDOJ) = DOL(I,LDO) + OH(I,LD7)
      GO TO 70
   72 NOH = - NOH
      LD7 = 7*(NOH - 1)
      LDO = ITOPOL(1,NOH)
      DO 73 I=1,3
   73 DOL(I,LDOJ) = DOL(I,LDO) - OH(I,LD7)
   70 CONTINUE
C
      IF(.NOT.LEQU) RETURN
      I1 = 6*(NH-1)                                                     0 4556
      I2 = 7*(NH-1)                                                     0 4557
      WRITE(NCT,1002) (I,((RH(J,L,I),J=1,3),L=1,3),I=1,I1)              0 4558
      WRITE(NCT,1003) (I,(UH(J,I),J=1,3),I=1,I2)                        0 4559
      WRITE(NCT,1004) (I,((PIN(J,L,I),J=1,3),L=1,3),I=1,NH)             0 4560
      WRITE(NCT,1005) (I,((HJE(J,L,I),J=1,3),L=1,3),I=1,NH)             0 4561
      WRITE(NCT,1006) (I,(UJE(J,I),J=1,3),I=1,NH)                       C 4562
      WRITE(NCT,1007) (I,((RP2(J,L,I),J=1,3),L=1,3),I=1,NH)             0 4563
      WRITE(NCT,1008) (I,((RP3(J,L,I),J=1,3),L=1,3),I=1,NH)             C 4564
 1002 FORMAT (1X,'  RH(',I2,') =',1P9D11.3)                             0 4565
 1003 FORMAT (1X,'  UH(',I2,') =',1P3D11.3)                             0 4566
 1004 FORMAT (1X,' PIN(',I2,') =',1P9D11.3)                             0 4567
 1005 FORMAT (1X,' HJE(',I2,') =',1P9D11.3)                             0 4568
 1006 FORMAT (1X,' UJE(',I2,') =',1P3D11.3)                             0 4569
 1007 FORMAT (1X,' RP2(',I2,') =',1P9D11.3)                             0 4570
 1008 FORMAT (1X,' RP3(',I2,') =',1P9D11.3)                             0 4571
      RETURN                                                            0 4572
C                                                                       0 4573
  999 WRITE (NCT,2001)                                                  0 4574
 2001 FORMAT (1H1,20HTOPOLOGY ERROR, ITOPOL)                            0 4575
      STOP                                                              0 4576
C                                                                       0 4577
      END                                                               0 4578
```

```
      SUBROUTINE EHGENR                                                 C 4579
C                                                          DEBUG #  45  C 4580
C                                                                       0 4581
C        SUBR TO GENERATE ELEMENTS WHICH DEFINE HINGE POINT             0 4582
C        KINEMATICS VOL 1,PAGE 36                                       0 4583
C                                                                       0 4584
      IMPLICIT REAL*8(A-H,O-Z)                                          0 4585
C                                                                       0 4586
            COMMON /GRDGRD/                                             0 4587
     *     RH(3,3,6,11),UH(3,10,15),HJE(3,3, 6),UJE(3, 6)               2 4588
            COMMON /HARDS /                                             0 4589
     *     RATM(3,12,15),SIGM(3,12,13),MATS(3,12,15),SIGS(3,12,15)      4 4590
            COMMON /MAXNUM/                                             0 4591
     *     NJMAX,NHMAX,NSPMAX,NMBMAX,NYMCD,NTJOSS,NRU,KT,KU             0 4592
            COMMON /NOPGRS/                                             0 4593
     *     ZHE,UNE,TWE,THE                                              C 4594
            COMMON /PINRP /                                             0 4595
     *     PIN(3,3, 6), RP2(3,3, 6), RP3(3,3, 6)                        13 4596
            COMMON /SPECIF/                                             0 4597
     *     DELTAH(6, 6),DELTAD(6, 6),AMU(2, 5),RM(3,3,30),RS(3,3,33),   16 4598
     *     UM(3,30),US(3,30),AMUS(2, 5),NPCA(5, 3),ITSMA(15),           17 4599
     *     NJ,NH,NSPT,NCMB,NDELTA,ITOPOL(2, 6),IROPLA( 6),IHDATA(7, 6), 18 4600
     *     LSEG(14),LENS(14),NU,NBETA,NLAM,NP3                          19 4601
C                                                                       C 4602
      DIMENSION W1(3,3),W2(3,3)                                         C 4603
C                                                                       0 4604
      DATA IST /0 /                                                     C 4605
      DATA NOT/0 /                                                      C 4606
      LOGICAL DEBUG,LE45                                                0 4607
      COMMON /LDEBUG/ DEBUG(120)                                        0 4608
      EQUIVALENCE (LE45,DEBUG(45))                                      0 4609
      IF(LEQU) WRITE (NOT,1000)
 1000 FORMAT (' SUBROUTINE UHGENR ')
C
      IF (IIST .EQ. 1) GO TO 100
      IIST = 1
      LR = 2*NFMAX - 1
      JR = 6 + NMOBOD
      DO 5 L=1,LR
      DO 5 I=1,6
      DO 5 J=1,JR
    5 UH(I,J,L) = ZRO
C
  100 DO 10 I=1,3
      IP3 = I + 3
      DO 10 J=1,3
      JP3 = J + 3
      UH(I,J,1) = PIN(I,J,1)
   10 UH(IP3,JP3,1) = ROL(I,J,1)
C
      DO 20 L=2,NM
      LQ = 2*L - 2
      LP = LQ + 1
C
C           LQ = 2,4,6,...
C           LP = 3,5,7,...
C           UH(LP) - EQUATION II-82, VOL 1
C           UH(LQ) = EQUATION II-83, VOL 1
C
      LR3 = 6*(L-2) + 3
      LR4 = LR3 + 1
      LR5 = LR3 + 2
      LU3 = 7*(L-2) + 3
      LU4 = LU3 + 1
      DO 25 I=1,3
      DO 25 J=1,3
      W1(J,I) = RM(I,J,LR3)
      W2(J,I) = RM(I,J,LR5)
   25 UH(J+3,I+3,LP) = -RM(I,J,LR4)
      CALL MULT3  (RM(1,1,LR5),W1,UH(4,4,LQ),3,3,3,3,3,6)
      CALL MULT3  (PIN(1,1,L),W1,UH(1,1,LQ),3,3,3,3,3,6)
      CALL MULT3  (W2,UH(4,4,LP),W1,3,3,3,3,6,3)
      CALL MULT3  (PIN(1,1,L),W1,UH(1,1,LP),3,3,3,3,3,6)
      CALL SKEWV3 (UM(1,LU3),W1,3,3)
      CALL SKEWV3 (UM(1,LU4),W2,3,3)
      CALL MULT3  (UH(4,4,LQ),W1,UH(4,1,LQ),3,3,3,6,3,6)
      CALL MULT3  (UH(4,4,LP),W2,UH(4,1,LP),3,3,3,6,3,6)
      NOBQ = ITOPOL(1,L)
      NOBP = ITOPOL(2,L)
      NRQ = IRFLEX(NOBQ)
      NRP = IRFLEX(NOBP)
C
C           IS Q-FRAME IN A RIGID BODY?
      IF (NRQ .EQ. 0) GO TO 30
      LHS = 2*L - 3
      CALL MULT3  (UH(1,1,LQ),SIGM(1,1,LHS),UH(1,7,LQ),3,3,NMQ,6,3,6)
      CALL MULT3  (UH(4,4,LQ),PATH(1,1,LHS),UH(4,7,LQ),3,3,NMQ,6,3,6)
C
C           IS P-FRAME IN A RIGID BODY?
   30 IF (NRP .EQ. 0) GO TO 20
      LHS = 2*L - 2
      CALL MULT3  (HH(1,1,L2),SIGH(1,1,LHS),DH(1,7,L2),3,3,NMP,6,3,6)     0 4670
      CALL MULT3  (HH(4,1,L2),HATH(1,1,LHS),DH(4,7,L2),3,3,NMP,6,3,6)     0 4671
   20 CONTINUE                                                            0 4672
C                                                                         0 4673
      IF(.NOT.LE45) RETURN                                                0 4674
      LHX = L*NH-1                                                        0 4675
      LHY = L*NMODES                                                      0 4676
      DO 40 L=1,LHX                                                       0 4677
      WRITE (NOUT,1001) L                                                 0 4678
 1001 FORMAT(' DH(',I2,') ',)                                             0 4679
   40 CALL WRITD (DH(1,1,L),6,LHY,6)                                      0 4680
      RETURN                                                              0 4681
      END                                                                 0 4682




      SUBROUTINE RETDS                                                    0 4683
C                                                                         0 4684
C                                                      DEBUG #  46        0 4685
C                                                                         0 4686
C        COMPUTES RS ROTATION TRANSFORMATION BETWEEN SENSOR AXIS SYSTEM   0 4686
C        AND BODY FIXED AXIS SYSTEM VOL II,PAGE 19                        0 4687
C        NOTHING DONE IF SENSOR ON RIGID BODY                             0 4688
      IMPLICIT REAL*8(A-H,O-Z)                                            0 4689
C                                                                         0 4690
            COMMON /HANDS /                                               0 4691
     *      HATF(3,12,10),SIGF(3,12,10),HATS(3,12,15),SIGS(3,12,15)       4 4692
            COMMON /SPECI /                                               0 4693
     *      BETA(6, 6),BETAD(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),      16 4694
     *      DH(3,30),DS(3,30),IMOD(5, 5),NAGA(5, 5),ITSM(15),            17 4695
     *      NB,NT,NSPT,NORMD,NDELTA,ITYPOL(2, 6),IHSPEX( 6),IHDATA( , 6), 16 4696
     *      LOCC(16),LENG(16),NO,NDELTA,NCAM,NE                          15 4697
            COMMON /VECTOR/                                               0 4698
     *      Y(250),YDT(250)                                              20 4699
C                                                                         0 4700
      DATA NOUT/6 /                                                       0 4701
      LOGICAL DEBUG,LEGO                                                  0 4702
      COMMON /LDEBUG/ DEBUG(120)                                          0 4703
      EQUIVALENCE (LEGO,DEBUG(46))                                        0 4704
      DIMENSION DF(3),AF(3),RF(3,3)                                       0 4705
                                                                          0 4706
      IF(LEGO) WRITE(NOUT,1000)                                           0 4707
      DO 10 L=1,NSPT                                                      0 4708
      NOD = ITSM(L)                                                       0 4709
      IF (IHSPEX(NOD) .EQ. 0) GO TO 10                                    0 4710
      LR1 = 2*L - 1                                                       0 4711
      LR2 = LR1 + 1                                                       0 4712
      LC = LOCC(NO+NOB)                                                   0 4713
      LF = LENG(NO+NOB)                                                   0 4714
      CALL MULT3  (HATS(1,1,L),Y(LC),DF,3,LF,1,3,1,1)                     0 4715
      CALL MULT3  (SIGS(1,1,L),Y(LC),AF,3,LF,1,3,1,1)                     0 4716
      CALL ROTTR (3,1,AF,RF,DUM,DUM)                                      0 4717
      CALL MULT3  (RF,RS(1,1,LR1),RS(1,1,LR2),3,3,3,3,3,3)                0 4718
      DO 15 I=1,3                                                         0 4719
   15 DS(I,LR2) = DS(I,LR1) + DF(I)                                       0 4720
C                                                                         0 4721
   10 CONTINUE                                                            0 4722
      IF(.NOT.LEGO) RETURN                                                0 4723
      II = 2*NSPT                                                       0 4724
      WRITE(NOT,1002)  (I,((RS(J,L,I),J=1,3),L=1,3),I=1,II)             0 4725
      WRITE(NOT,1003)  (I,(DS(J,I),J=1,3),I=1,II)                       0 4726
 1000 FORMAT (' SUBROUTINE RGIDS ENTERED')                              0 4727
 1002 FORMAT (1X,'  RS(',I2,') =',1P9D11.3)                             0 4728
 1003 FORMAT (1X,'  DS(',I2,') =',1P3D11.3)                             0 4729
C                                                                       0 4730
      RETURN                                                            0 4731
      END                                                               0 4732




      SUBROUTINE DSGENR                                                 0 4733
C                                                        DEBUG #  47    0 4734
C                                                                       0 4735
C        GET KINEMATICAL COEFFICIENTS FOR ALL SENSOR POINTS             0 4736
C        SEE VOL II,PAGE 17                                             0 4737
C        EQUATIONS ANALOGOUS TO THOSE PROGRAMMED IN DHGENR              0 4738
      IMPLICIT REAL*8(A-H,O-Z)                                          0 4739
C                                                                       0 4740
            COMMON /HMGSHD/                                             0 4741
     *     BH(6,18,11),BS(6,18,15),POL(3,3, 6),DDC(3, 6)                2 4742
            COMMON /HANDS /                                             0 4743
     *     HATH(3,12,10),SIGH(3,12,10),HATS(3,12,15),SIGS(3,12,15)      4 4744
            COMMON /MAXNUM/                                             0 4745
     *     NGMAX,NHMAX,NSPMAX,NMHMAX,NMHCOD,NMODOD,NXD,NY,ND            0 4746
            COMMON /NUMBRS/                                             0 4747
     *     ZRO,ONE,TWO,THRS                                             0 4748
            COMMON /SPECIF/                                             0 4749
     *     DETAH(6, 6),DETAHD(6, 6),ARC(2, 5),RH(3,3,30),RS(3,3,30),   16 4750
     *     DH(3,30),DS(3,30),IMO(3, 5),NMCH(5, 3),IFTSMH(15),          17 4751
     *     NG,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 6),IHGFLX( 6),IHDATA(7, 6), 18 4752
     *     LOCC(14),LENG(14),NO,NDETA,NLAM,NEQ                         19 4753
C                                                                       C 4754
      DIMENSION W(3,3)                                                  0 4755
C                                                                       C 4756
      DATA IIST / 0 /                                                   0 4757
      DATA NOT/6/                                                       0 4758
      LOGICAL DEBUG,LEQU                                                0 4759
      COMMON /EDEBUG/ DEBUG(100)                                        0 4760
      EQUIVALENCE (LEQU,DEBUG(47))                                      C 4761
C                                                                       0 4762
      IF(LEQU) WRITE(NOT,1000)                                          0 4763
 1000 FORMAT (' SUBROUTINE DSGENR')                                     0 4764
      IF (IIST .EQ. 1) GO TO 20                                         0 4765
C                                                                       0 4766
      JR = 6 * NMODED                                                   0 4767
      DO 5 L=1,NSPT                                                     0 4768
      DO 5 I=1,6                                                        0 4769
      DO 5 J=1,JR                                                       0 4770
    5 BS(I,J,L) = ZRO                                                   0 4771
C                                                                       0 4772
   20 DO 10 L=1,NSPT                                                    0 4773
      NOB = IFTSMH(L)                                                   0 4774
      LE = IHGFLX(NOB)                                                  0 4775
      IF (LE .EQ. 0  .AND. IIST .EQ. 1) GO TO 10                        0 4776
      LH2 = 2*L                                                         0 4777
      DO 15 I=1,3                                                        0 4778
      IP3 = I + 3                                                        C 4779
      DO 15 J=1,3                                                        0 4780
      JP3 = J + 3                                                        C 4781
      ES(J,I,L) = RS(I,J,LR2)                                            0 4782
   15 ES(JP3,IP3,L) = RS(I,J,LR2)                                        0 4783
      CALL SKEWV3 (DS(1,LR2),W,3,3)                                      C 4784
      CALL MULTS  (DS(1,1,L),W,BS(4,1,L),3,3,3,6,3,6)                    0 4785
      IF (LL .EQ. 0) GO TO 10                                            0.4786
      CALL MULTS  (DS(1,1,L),SIGS(1,1,L),BS(1,7,L),3,3,LL,6,3,6)         0 4787
      CALL MULTS  (DS(1,1,L),HATS(1,1,L),BS(4,7,L),3,3,LL,6,3,6)         0 4788
   10 CONTINUE                                                           0 4789
C                                                                        C 4790
      IIST = 1                                                           0 4791
      IF(.NOT.LLOG) RETURN                                               0 4792
      LMY = 6+NMODE                                                      C 4793
      DO 25 L=1,NSPT                                                     0 4794
      WRITE (NOT,1001)L                                                  0 4795
 1001 FORMAT (' BS(',I2,') IS ')                                         C 4796
   25 CALL WRITE (BS(1,1,L),6,LMY,6)                                     0 4797
      RETURN                                                             0 4798
      END                                                                0 4799
```

```
      SUBROUTINE GETBMB                                                  0 4800
C                                                       DEBUG #   48     C 4801
      IMPLICIT REAL*8(A-H,O-Z)                                           C 4802
C                                                                        C 4803
            COMMON /AMODE /                                              C 4804
     *      AMODE(22, 3),DM(33,13)                                       1 4805
            COMMON /BMODE3/                                              0 4806
     *      BM(6,6,11),BS(6,18,30),RCL(3,3, 6),DDL(3, 6)                 2 4807
            COMMON /EVCONS/                                              0 4808
     *      E(6, 6)                                                      9 4809
            COMMON /MAXNUM/                                              0 4810
     *      NMAX,NOMAX,NSPMAX,N4MAX,NMFDD,NMDSSS,N4J,NT,KD               0 4811
            COMMON /NUMBRS/                                              C 4812
     *      ZRO,ONE,TWO,THRE                                             C 4813
            COMMON /SPECIF/                                              0 4814
     *      DLTAM(6, 5),DLTAMD(6, 6),AMC(2, 5),RR(3,3,30),RS(3,3,30),    16 4815
     *      DS(3,30),ES(6,30),IMOD(3, 5),NMOD(6, 5),ITSMA(15),           17 4816
     *      NS,NMOD,NSPT,NOMDS,NDELTA,ITCHOL(2, 6),IROPLA( 6),INDATA(7, 6), 18 4817
     *      LSCCT(4),LLNG(14),NU,NDLTA,NLAM,NL3                          19 4818
C                                                                        0 4819
      DIMENSION BMS(30,30),BM(6,6,11), ITOP( 6, 6),W(18)                 82 4820
      EQUIVALENCE (W(1),BMS(1)),  (W(1297),BM(1))                        77 4821
      DATA NU/6 /                                                        0 4822
      LOGICAL DEBUG,LLOG                                                 0 4823
      COMMON /LDEBUG/ DEBUG(120)                                         0 4824
      EQUIVALENCE (LLOG,DEBUG(48))                                       0 4825
C                                                                        C 4826
      DATA IIST / 0 /                                                    C 4827
C                                                                        0 4828
      IF(LLOG) WRITE(NOT,1000)                                           0 4829
 1000 FORMAT (' SUBROUTINE GETBMB')                                      0 4830
      IF (IIST .EQ. 1) GO TO 100                                         0 4831
      IIST = 1                                                          0 4832
      DO 5 I=1,NH                                                       0 4833
      DO 5 J=1,NB                                                       0 4834
    5 ITCP(I,J) = 0                                                     0 4835
      ITCP(1,1) = 1                                                     0 4836
      DO 3 I=1,6                                                        0 4837
      DO 3 J=1,NH                                                       0 4838
    3 IV(I,J) = 0                                                       0 4839
C                                                                       0 4840
C         LOAD IV ARRAY, USED TO SIGNAL CONSTRAINT CONDITION            0 4841
      IC = 0                                                            0 4842
      DO 7 J=1,NH                                                       0 4843
      DO 7 I=1,6                                                        0 4844
      IP1 = I + 1                                                       0 4845
      IF (IHDATA(IP1,J) .EQ. 0) GO TO 7                                 0 4846
      IC = IC + 1                                                       0 4847
      IV(I,J) = IC                                                      0 4848
    7 CONTINUE                                                          0 4849
C                                                                       0 4850
C         ITCP ARRAY USED TO FIND STORAGE LOCATION IN OM ARRAY FOR      0 4851
C         PARTICULAR VELOCITY TRANSFORMATIONS. EQUATIONS II-82,83 VOL 1 0 4852
C         ITCP(L,NQ) = STORAGE LOCATION FOR Q FRAME AT HINGE L BODY NQ  0 4853
C         ITCP(L,NP) = STORAGE LOCATION FOR P FRAME AT HINGE L BODY NP  0 4854
      DO 10 L=2,NH                                                      0 4855
      LQ = 2*L - 2                                                      0 4856
      LP = LQ + 1                                                       0 4857
      NQ = ITOPOL(1,L)                                                  0 4858
      NP = ITOPOL(2,L)                                                  0 4859
      ITCP(L,NQ) = LQ                                                   0 4860
   10 ITCP(L,NP) = LP                                                   0 4861
C                                                                       0 4862
C         SELECTIVE ROW MULTIPLICATIONS USED TO COMPUTE                 0 4863
C                    OMT = D * M**(-1) * D**T                           0 4864
C         THIS MATRIX USED IN EQUATION II-6 TO FIND LAGRANGE MULTIPLIERS 0 4865
C         SEE PAGE 37,VOL 1                                             0 4866
CC        SAVE EM = D * M**(-1) NEEDED FOR EQUATION II-6 EVALUATION     0 4867
  100 LCG = 6 * INOFLX(1)                                               0 4868
      LMG = LENG(1)                                                     0 4869
      CALL MLTSM  (OM,AMO,DM,LCG,LMG,1,IV,KMO)                          0 4870
C                                                                       0 4871
      DO 20 L=2,NH                                                      0 4872
      NOBQ = ITOPOL(1,L)                                                0 4873
      NOBP = ITOPOL(2,L)                                                0 4874
      LCQ = INOFLX(NOBQ) * 6                                            0 4875
      LCP = INOFLX(NOBP) * 6                                            0 4876
      LMQ = LENG(NOBQ)                                                  0 4877
      LMP = LENG(NOBP)                                                  0 4878
      LQ = 2*L - 2                                                      0 4879
      LP = LQ + 1                                                       0 4880
      CALL MLTSM (OM(1,1,LQ),AMO(1,1,NOBQ),DM(1,1,LQ),LCQ,LMQ,L,IV,KMO) 0 4881
      CALL MLTSM (OM(1,1,LP),AMO(1,1,NOBP),DM(1,1,LP),LCP,LMP,L,IV,KMO) 0 4882
   20 CONTINUE                                                          0 4883
C                                                                       0 4884
      DO 25 I=1,NLAM                                                    0 4885
      DO 25 J=1,NLAM                                                    0 4886
   25 OMT(I,J) = ZRO                                                    0 4887
C                                                                       0 4888
      DO 30 N=1,NB                                                      0 4889
      LF = INOFLX(N) * 6                                                0 4890
      DO 30 L=1,NH                                                      0 4891
      IF (ITOP(L,N) .EQ. 0) GO TO 35                                     0 4892
      DO 40 I=L,NN                                                       0 4893
      IF (ITOP(I,N) .EQ. 0) GO TO 40                                     0 4894
      LOW = ITOP(L,N)                                                    0 4895
      LIT = ITOP(I,N)                                                    0 4896
      DO 50 M=1,6                                                        0 4897
      NE = IV(M,L)                                                       0 4898
      IF (NE .EQ. 0) GO TO 50                                            0 4899
      DO 55 K=1,LL                                                       0 4900
   55 WR(K) = CH(M,K,LOW)                                                0 4901
      DO 60 J=1,6                                                        0 4902
      NI = IV(J,I)                                                       0 4903
      IF (NI .LT. NE) GO TO 60                                           0 4904
      S = ZRO                                                            0 4905
      DO 65 K=1,LL                                                       0 4906
   65 S = S + WR(K)*CH(J,K,LIT)                                          0 4907
      EMU(NE,NI) = EMU(NE,NI) + S                                        0 4908
   60 CONTINUE                                                           0 4909
   50 CONTINUE                                                           0 4910
   40 CONTINUE                                                           0 4911
   35 CONTINUE                                                           0 4912
   30 CONTINUE                                                           0 4913
C                                                                        0 4914
      IF(.NOT.LEGO) RETURN                                               0 4915
      WRITE (NLT,1001)                                                   0 4916
 1001 FORMAT (' THE EMU ARRAY IS ')                                      0 4917
      LDV = 6*NMMAX                                                      0 4918
      CALL WRITE(EMU,NEAM,NEAM,LDV)                                      0 4919
      RETURN                                                             0 4920
      END                                                                0 4921
```

```
      SUBROUTINE MUP                                                     0 4922
C                                                     DEBUG #  49        0 4923
C                                                                        0 4924
C        GENERATE OPERATION DEPENDENT MASS MATRIX CONTAINED IN           0 4925
C         EQUATION II-80 VOL I VIA EQUATIONS II-67,63,69                 0 4926
      IMPLICIT REAL*8(A-H,O-Z)                                           0 4927
C                                                                        0 4928
            COMMON /AMUDM /                                              0 4929
     *     AMU(22,22, 6),DM(30,11,3)                                     1 4930
            COMMON /DHDSHD/                                              0 4931
     *     DH(6,10,11),DS(6,10,12),PDC(3,3, 6),DDC(3, 6)                 2 4932
            COMMON /DDSAVE/                                              0 4933
     *     DDS(12,4, 6)                                                  3 4934
            COMMON /INTGRL/                                              0 4935
     *     A4(17, 5),ACOF(9,12, 6),BCOF(6,12, 6),                        5 4936
     *     COF11(12,12, 6),COF22(12,12, 6),COF33(12,12, 6),AK(12,12, 6), 6 4937
     *     COF12(12,12, 6),COF13(12,12, 6),COF23(12,12, 6),AU(12,12, 6), 7 4938
     *     COFXY(12,12, 6),COFXZ(12,12, 6),COFYZ(12,12, 6)               8 4939
            COMMON /MAXNUM/                                              0 4940
     *     NUMAX,NMMAX,NSPMAX,NMWMAX,NMB[illegible],NMDDDD,KMU,KY,KU      0 4941
            COMMON /NUMBRS/                                              0 4942
     *     ZRO,ONE,TWO,THRE                                              0 4943
            COMMON /SPECIF/                                              0 4944
     *     DETAM(6, 6),DETAMD(6, 6),AVG(2, 3),RH(3,3,33),RS(3,3,33),     16 4945
     *      CM(3,35),CS(3,35),IMG(3, 5),NMCA(6, 3),IFISMA(15),              17 4946
     *      NS,NM,NSPT,NGRMU,NDELTA,ITCPOL(2, 5),INGFLX( 6),INDATA(7, 6),  18 4947
     *      LOCC(14),LENG(14),ND,NDLTA,NLAM,NCQ                             15 4948
              COMMON /VECTOR/                                                0 4949
     *      Y(250),YDT(250)                                                 20 4950
C                                                                            0 4951
      DIMENSION HM(3,12),CM(12,3),VM(9),AV(6)                               83 4952
      DATA NOT/ 6 /                                                          0 4953
      LOGICAL DEBUG,LEDO                                                     0 4954
      COMMON /DEBUG/ DEBUG(20)                                               0 4955
      EQUIVALENCE (LEDO,DEBUG(4))                                            0 4956
      IF (LEDO) WRITE(NOT,1000)                                              0 4957
 1000 FORMAT (' SUBROUTINE AGENT')                                           0 4958
C                                                                            0 4959
      KM = NMODES                                                            0 4960
      DO 10 N=1,ND                                                           0 4961
      LO = LOCC(ND+N)                                                        0 4962
      LE = LENG(ND+N)                                                        0 4963
      KNT = 0                                                                0 4964
      NPO = 6 + LE                                                           0 4965
      DO 12 I=1,NPO                                                          0 4966
      DO 12 J=1,NPO                                                          0 4967
      KNT = KNT + 1                                                          0 4968
   12 AMU(I,J,N) = AM(KNT,N)                                                 0 4969
      IF (LE .EQ. 0) GO TO 50                                                0 4970
      CALL MULT3  (DCOF(1,1,N),Y(LO),VM,6,LE,1,6,1,1)                        0 4971
      CALL MULT3  (COF11(1,1,N),Y(LO),CM(1,1),LE,LE,1,KM,1,KM)               0 4972
      CALL MULT3  (COF22(1,1,N),Y(LO),CM(1,2),LE,LE,1,KM,1,KM)               0 4973
      CALL MULT3  (COF33(1,1,N),Y(LO),CM(1,3),LE,LE,1,KM,1,KM)               0 4974
      CALL MULT3  (Y(LO),COF12(1,1,N),HM(1,1),1,LE,LE,1,KM,3)                0 4975
      CALL MULT3  (Y(LO),COF13(1,1,N),HM(2,1),1,LE,LE,1,KM,3)                0 4976
      CALL MULT3  (Y(LO),COF23(1,1,N),HM(3,1),1,LE,LE,1,KM,3)                0 4977
C                                                                            0 4978
C     FILL OF DATA FOR GRAVITY GRADIENT EFFECTS ON ELASTIC COORDINATES       0 4979
C                                                                            0 4980
      DO 16 J=1,LE                                                           0 4981
      CGS(J,1,N) = CM(J,1)                                                   0 4982
      CGS(J,2,N) = CM(J,2)                                                   0 4983
      CGS(J,3,N) = CM(J,3)                                                   0 4984
      CGS(J,7,N) = HM(1,J)                                                   0 4985
      CGS(J,8,N) = HM(2,J)                                                   0 4986
   16 CGS(J,9,N) = HM(3,J)                                                   0 4987
C                                                                            0 4988
      CALL MULT3  (Y(LO),CM,AV(1),1,LE,3,1,KM,1)                             0 4989
      CALL MULT3  (HM,Y(LO),AV(4),3,LE,1,3,1,1)                              0 4990
      DO 15 I=1,3                                                            0 4991
   15 AMU(I,I,N) = AMU(I,I,N) + 2.0*VM(I) + AV(I)                            0 4992
      AMU(1,2,N) = AMU(1,2,N) - VM(4) - AV(4)                                0 4993
      AMU(1,3,N) = AMU(1,3,N) - VM(5) - AV(5)                                0 4994
      AMU(2,3,N) = AMU(2,3,N) - VM(6) - AV(6)                                0 4995
      CALL MULT3 (ACOF(1,1,N),Y(LO),VM,9,LE,1,9,1,1)                         0 4996
      DO 17 J=1,3                                                            0 4997
      JP3 = J + 3                                                            0 4998
      DO 17 I=1,3                                                            0 4999
      IJ = J + 3*(I-1)                                                       0 5000
   17 AMU(I,JP3,N) = AMU(I,JP3,N) + VM(IJ)                                   0 5001
      CALL MULTAD (Y(LO),COFYZ(1,1,N),AMU(1,7,N),1,LE,LE,1,KM,KMU)           0 5002
      CALL MULTAD (Y(LO),COFXZ(1,1,N),AMU(2,7,N),1,LE,LE,1,KM,KMU)           0 5003
      CALL MULTAD (Y(LO),COFXY(1,1,N),AMU(3,7,N),1,LE,LE,1,KM,KMU)           0 5004
C                                                                            0 5005
```

```
      CALL MULT3  (COF12(1,1,N),Y(LO),CW(1,1),LC,LC,1,KM,1,KM)
      CALL MULT3  (COF13(1,1,N),Y(LO),CW(1,2),LC,LC,1,KM,1,KM)
      CALL MULT3  (COF23(1,1,N),Y(LO),CW(1,3),LC,LC,1,K4,1,KM)
C
C  FINISH FEEDING OFF GRAVITY GRADIENT DATA
C
      DO 28 J=1,LC
      GGS(J,4,N) = CW(J,1)
      GGS(J,5,N) = CW(J,2)
   28 GGS(J,6,N) = CW(J,3)
C
C
C          ACCOUNT FOR ALL ACTIVE WHEELS IN BODY N
   30 NMWVS = NMCW(2,N)
      IF (NMWVS .EQ. 0) GO TO 110
      NMW = NMCW(1,N)
      LCBS = 6 * LC
      NV = 0
      J1 = LCBS + 1
      J2 = LCBS + NMWVS
      DO 70 L=1,NMW
      LP2 = L + 2
      NUMW = NMCW(LP2,N)
      IF (IMW(3,NUMW) .EQ. 3) GO TO 70
C          SEE VOL 1,PAGE 47,GET ALL HCT 1,1 TERM IN EQ 11-139
C
      NV = NV + 1
      LV = 6 * LC + NV
      NUSP = IMW(1,NUMW)
      NA = IMW(2,NUMW)
      AJS = AMW(2,NUMW)
      DO 75 J=1,LCBS
   75 AMO(J,LV,N) = AJS*US(NA,J,NUSP)
      DO 76 J=J1,J2
   76 AMO(LV,J,N) = ZRO
      AMO(LV,LV,N) = AJS
   70 CONTINUE
C
  110 LCQ = LCNO(N)
      DO 77 I=1,LCQ
      DO 77 J=1,LCQ
   77 AMO(J,I,N) = AMO(I,J,N)
C
      IF(LCP.LE.0) GO TO 20
      WRITE (NCT,1001) N
 1001 FORMAT (' AMO, MASS MATRIX FOR BODY',I3)
      LQ2 = 6*NNOBOD+NMWLCQ
      CALL WRITES(AMO(1,1,N),LCQ,LCQ,LC*6)
      IF(LCP.LE.0) GO TO 20
      WRITE (NCT,1002) N
 1002 FORMAT (' GGS, GRAVITY GRADIENT EFFECTS ON MODAL COORDINATES OF BO
     *DY',I3)
      CALL WRITES(GGS(1,1,N),LC,6,MAXLC)
   20 CONTINUE
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE GRVGRD (GGV)
C                                                         DEBUG #  50
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GGV(1)
C
            COMMON /AMODB /
     *     AMU(22,22, 6),DB(36,113)
            COMMON /BMDSHD/
     *     BH(6,18,11),BS(6,18,15),ROL(3,3, 6),DJL(3, 6)
            COMMON /GGDATA/
     *     GAMGI(3),GMAG,RCMAG
            COMMON /GGSAVE/
     *     GGS(12,9, 6)
            COMMON /INTGRL/
     *     AM(171, 6),ACCF(9,12, 6),BCOF(6,12, 6),
     *     COF11(12,12, 6),COF22(12,12, 6),COF33(12,12, 6),AK(12,12, 6),
     *     COF12(12,12, 6),COF13(12,12, 6),COF23(12,12, 6),AD(12,12, 6),
     *     COFXY(12,12, 6),COFXZ(12,12, 6),COFYZ(12,12, 6)
            COMMON /MAXNOS/
     *     NBMAX,NHMAX,NSPMAX,NMMAX,NMMEGD,NMDJJD,NAJ,NY,ND
            COMMON /NUMBRS/
     *     ZRO,ONE,TWO,THRE
            COMMON /SPECIF/
     *     DETAH(6, 6),DETAHD(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),
     *     DH(3,35),DS(3,30),IMO(3, 5),NMOD(6, 6),IFISMO(15),
     *     NB,NH,NSPI,NMMAC,NDELTA,ITCPOL(2, 6),IRGFLA( 6),IMDATA(7, 6),
     *     LOCO(14),LENG(14),ND,NDLTA,NLAM,NEQ
            COMMON /VECTOR/
     *     Y(250),YDT(250)
C
      DIMENSION GAMGL(3),V(3)
      DATA NOT / 6/
      LOGICAL DEBUG,LEQU
      COMMON /LDEBUG/ DEBUG(20)
      EQUIVALENCE (LEQU,DEBUG(50))
      IF(LEQU) WRITE(NOT,1000)
 1000 FORMAT(' SUBROUTINE GRVGRD ')
C         ****   CAUTION WHEN APPLYING GRAVITY GRADIENT CAPABILITY
C         *****           IT IS NOT APPLICABLE TO ALL PROBLEMS
C         *****           UPDATE UNDER DEVELOPMENT, CALL DOOLEY
C        FROM DYNSIO:
C              GAMGI - UNIT VECTOR IN DIRECTION OF GRAVITY FIELD INERTIAL
C                      COORDINATES
C              GMAG  - LOCAL GRAVITATIONAL ACCELERATION
C              RCMAG - RADIUS VECTOR TO VICINITY OF SPACECRAFT
C                      FROM GRAVITY SOURCE
      IF (GMAG .EQ. ZRO) RETURN
      IF (RCMAG .LE. ONE) GO TO 999
      DO 10 N=1,NB
      LOO = LOCO(N)
      LG = LOCO(N+ND)
      LL = LENG(N+ND)
C
C        EVALUATE EQUATION II-125 FIRST THREE ELEMENTS OF THE
C        CONTRIBUTATION TO THE FORCE VECTOR DUE TO GRAVITY GRADIENT
C        GAMGL - UNIT GRAVITY DIRECTION VECTOR LOCAL COORDINATES
      CALL MULT3  (GAMGI,ROL(1,1,N),GAMGL,1,3,3,1,3,1)
C        AMU(4,1,N) STARTING LOCATION FOR FIRST MASS MOMENT COEFFICIENTS
      CALL MULT3  (AMU(4,1,N),GAMGL,GGV(LOO),3,3,1,NAJ,1,1)
      CALL MULT3 (AMU(1,1,N),GAMGL,V,3,3,1,NMU,1,1)
      GOV(LOC+1) = XMAG*(GOV(LOC+1)                                     O 5124
     * + THC3*(GAMMA(2)*V(3) - GAMMA(3)*V(2))/RCMAG)                    C 5125
      GOV(LOC+2) = XMAG*(GOV(LOC+2)                                     C 5126
     * + THC3*(GAMMA(3)*V(1) - GAMMA(1)*V(3))/RCMAG)                    O 5127
      GOV(LOC+3) = XMAG*(GOV(LOC+3)                                     O 5128
     * + THC3*(GAMMA(1)*V(2) - GAMMA(2)*V(1))/RCMAG)                    C 5129
C                                                                       O 5130
C         GET SECOND THREE TERMS DEFINED BY EQUATION II-126             C 5131
C         WITH CUT MASS MOMENT COEFFICIENTS                             O 5132
      V(1) = AMO(3,2,N)                                                 O 5133
      V(2) = AMO(1,3,N)                                                 O 5134
      V(3) = AMO(2,1,N)                                                 O 5135
      [illegible]                                                       C 5136
      GCY = -GMAG*AMO(4,4,N)                                            O 5137
      GCH = -GMAG/RCMAG                                                 O 5138
      LOC2 = LOC + 3                                                    O 5139
      DO 20 I=1,3                                                       O 5139
      GOV(LOC2+I) = GCY*GAMMA(I) + GCH*V(I)                             C 5140
   20 S = S + GAMMA(I)*V(I)                                             O 5141
      S = -THC3*S*GCH                                                   O 5142
      DO 25 I=1,3                                                       O 5143
   25 GOV(LOC2+I) = GOV(LOC2+I) + S*GAMMA(I)                            C 5144
C                                                                       O 5145
C         GET THE FORCE DUE TO GRAVITY ACTING ON THE DEFORMATION MODES  O 5146
C         (C EQUATION II-127                                           C 5147
      IF (LG .EQ. 0) GO TO 15                                           O 5148
      LOCN = LOC + 6                                                    O 5149
      V(1) = GAMMA(1)*RCMAG                                             O 5150
      V(2) = GAMMA(2)*RCMAG                                             O 5151
      V(3) = GAMMA(3)*RCMAG                                             C 5152
      CALL MULTB (V,AMO(4,7,N),GOV(LOCN),1,3,LG,1,N40,1)                O 5153
      CALL MULTAB (AMO(7,7,N),V(3),GOV(LOCN),LG,LG,1,N40,1,1)           O 5154
      LOCS = LOC + 6                                                    O 5155
      DO 10 J=1,LG                                                      O 5156
      GOV(LOCS+J) = GCH*GOV(LOCS+J)                                     O 5157
     *          + GCY*(HCOF(1,J,N) + HCOF(2,J,N) + HCOF(3,J,N))/TAU     O 5158
   10 CONTINUE                                                          O 5159
C                                                                       C 5160
      V(1) = ONE - TAU*GAMMA(1)*GAMMA(1)                                O 5161
      V(2) = ONE - TAU*GAMMA(2)*GAMMA(2)                                O 5162
      V(3) = ONE - TAU*GAMMA(3)*GAMMA(3)                                O 5163
      S = THC3*GMAG/(TAU*RCMAG)                                         O 5164
C                                                                       O 5165
      DO 40 J=1,LG                                                      C 5166
      GOV(LOCS+J) = GOV(LOCS+J)                                         C 5167
     *   + S* (V(1)*(HCOF(1,J,N) + GGS(J,1,N))                          O 5168
     *   +        V(2)*(HCOF(2,J,N) + GGS(J,2,N))                       O 5169
     *   +        V(3)*(HCOF(3,J,N) + GGS(J,3,N))                       O 5170
     * + TAU*GAMMA(1)*GAMMA(2)*(HCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))  O 5171
     * + TAU*GAMMA(1)*GAMMA(3)*(HCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))  C 5172
     * + TAU*GAMMA(2)*GAMMA(3)*(HCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))) O 5173
   40 CONTINUE                                                          O 5174
C                                                                       O 5175
      IF(.NOT.LEG) GO TO 15                                             O 5176
      WRITE(KOUT,1031) N                                                C 5177
 1031 FORMAT (' CONTRIBUTION TO FORCE VECTOR FROM GRAVITY GRADIENT EFFEC O 5178
     *TS ON BODY',I3)                                                   O 5179
      L = LOCS+1                                                        O 5180
      CALL WRITEM(GOV(L),1,LG,1)                                        C 5181
   15 CONTINUE                                                          O 5182
                                                                        O 5183
      RETURN                                                            0 5184
C                                                                       0 5185
  999 WRITE (NOT,2001)                                                  0 5186
 2001 FORMAT (1H1,29HNCMAS - 3., SUBROUTINE GRVS(J)                     0 5187
      STOP                                                              0 5188
      END                                                               0 5189




      SUBROUTINE BDOTGP (L,BDTG,BDTP)                                   0 5190
C                                                          DEBUG #  51  0 5191
      IMPLICIT REAL*8(A-H,O-Z)                                          0 5192
C                                                                       0 5193
C        GET LOWER CASE B DOT MATRIX CALLED FOR IN EQ. II-5             0 5194
C        REFER TO APPENDIX C VOL I FOR THEORY ALSO PAGES 36-38          0 5195
C        EXIT WITH THE B DOT MATRIX TO BE USED IN EQ. II-6 ASSOCIATED   0 5196
C        WITH BODY L                                                    0 5197
C                                                                       0 5198
       DIMENSION BDTG(6,1),BDTP(6,1)                                    0 5199
C                                                                       0 5200
              COMMON /BMDSHJ/                                           0 5201
     *        BH(6,15,11),BS(6,15,15),ROL(3,3, 6),JJL(3, 6)             2 5202
              COMMON /HARDS /                                           0 5203
     *        HATH(3,12,10),SIGH(3,12,10),HATS(3,12,15),SIGS(3,12,15)   4 5204
              COMMON /NUMBRS/                                           0 5205
     *        ZRO,ONE,TWO,THRE                                          0 5206
              COMMON /PINRP /                                           0 5207
     *        RIN(3,3, 6), RP2(3,3, 6), RP3(3,3, 6)                    13 5208
              COMMON /SPECIF/                                           0 5209
     *        BETAM(6, 6),BETAMD(6, 6),AMD(2, 5),RM(3,3,30),RS(3,3,30), 16 5210
     *        JH(3,30),JS(3,30),IMD(3, 5),NMCW(6, 5),ITSMM(15),        17 5211
     *        NB,NH,NSPT,NORMB,NDELTA,ITCPOL(2, 5),IROPLA( 6),IRDATA(7, 6), 18 5212
     *        LOCC(14),LENDC(14),NU,NDELTA,NCAM,NFJ                    19 5213
              COMMON /VECTOR/                                           0 5214
     *        Y(250),YDT(250)                                          20 5215
C                                                                       0 5216
      DIMENSION PINCT(3,3),RADGN(3),SKDGN(3),RDPM(3),SKDPM(3),ROP(3,3), 0 5217
     *        RPM(3,3),RGN(3,3),RFN(3,3),RQN(3,3),DRPM(3,3),DRQP(3,3),  0 5218
     *        DRGN(3,3),DRGM(3,3),SNG(3,3),SMP(3,3),WGN(3),WPM(3),WPQP(3), 0 5219
     *        WSK(3,3), VEC(3)                                          0 5220
              COMMON /MAXMUM/                                           0 5221
     *        NBMAX,NHMAX,NSPMAX,NMWMAX,NMWHOL,NMDJJJ,K4J,KY,KU         0 5222
      DATA NOT/6 /                                                      0 5223
      LOGICAL DEBUG,LEGO                                                0 5224
      COMMON /LDEBUG/ DEBUG(120)                                        0 5225
      EQUIVALENCE (LEGO,DEBUG(51))                                      0 5226
      IF(LEGO) WRITE(NOT,1000)                                          0 5227
 1000 FORMAT (' SUBROUTINE BDOTGP ')                                    0 5228
C                                                                       0 5229
      DO 3 I=1,6                                                        0 5230
      DO 3 J=1,6                                                        0 5231
      BDTG(I,J) = ZRO                                                   0 5232
    3 BDTP(I,J) = ZRO                                                   0 5233
      IF (L .EQ. 1) GO TO 100                                           0 5234
C                                                                       0 5235
CC    GET D/DT(PI(INVERSE))                                             0 5236
      F2DT = BETAMD(2,L)                                                0 5237
```

CONTENTS

Page

PREFACE . . . . iii 1/A5

I. PROGRAM SYSTEM OVERVIEW 1 1/A7

- A. Introduction 2 1/A8
- B. Simulation Overview and Nomenclature 3 1/A9
- C. General Use Information 8 1/A14

II. PROGRAM SEGMENTATION 9 1/B1

III. DELINEATION OF USER-SUPPLIED MODULES 10 1/B2

- A. Logic Flow 11 1/B3
- B. User-Supplied Module Specifics 12 1/B4
- C. Selected Common-Block Information 17 1/B9

IV. PROGRAM INPUTS 19 1/B11

- A. Subroutines READ and READIM 19 1/B11
- B. Input Data Stream 20 1/B12

V. PROGRAM OUTPUTS 20 1/B12

VI. AUXILIARY PROGRAMS 33 1/C11

- A. REDIM—The Redimension Program 33 1/C11
- B. Support Programs for the DISCOS Flexible-Body Capability 42 1/D6

APPENDIX A—INPUT/OUTPUT SUBROUTINE EXPLANATIONS A-1 1/F12

APPENDIX B—SUPPORT PROGRAMS FOR THE DISCOS FLEXIBLE-BODY CAPABILITY B-1 2/A5

```
      B3DT = BETAHD(3,L)
      DO 5 I=1,3
      DO 5 J=1,3
    5 PINDT(I,J) = B2DT*RP2(I,J,L) + B3DT*RP3(I,J,L)
C
      NOBQ = ITOPOL(1,L)
      NOBP = ITOPOL(2,L)
      LOQ = LOCU(NOBQ) + 6
      LOP = LOCU(NOBP) + 6
C        THE Q-FRAME IS IN BODY NOBQ IT HAS LEQ FLEXIBLE BODY MODES
C        THE P-FRAME IS IN BODY NOBP IT HAS LEP FLEXIBLE BODY MODES
      LEQ = IRGFLX(NOBQ)
      LEP = IRGFLX(NOBP)
      LHSQ = 2*L - 3
      LHSP = LHSQ + 1
      IF (LEQ .EQ. 0) GO TO 10
      CALL MULT3  (HATH(1,1,LHSQ),Y(LOQ),HXDQN,3,LEQ,1,3,1,1)
      CALL MULT3  (SIGH(1,1,LHSQ),Y(LOQ),SXDQN,3,LEQ,1,3,1,1)
   10 IF (LEP .EQ. 0) GO TO 20
      CALL MULT3  (HATH(1,1,LHSP),Y(LOP),HXDPM,3,LEP,1,3,1,1)
      CALL MULT3  (SIGH(1,1,LHSP),Y(LOP),SXDPM,3,LEP,1,3,1,1)
   20 LRNQ = 6*(L-2) + 3
      LRMP = LRNQ + 1
      LRPQ = LRNQ + 2
      LDNQ = 7*(L-2) + 3
      LDMP = LDNQ + 1
      DO 15 I=1,3
      DO 15 J=1,3
      RQP(I,J) = RH(J,I,LRPQ)
      RQN(I,J) = RH(J,I,LRNQ)
   15 RPM(I,J) = RH(J,I,LRMP)
      CALL MULT3  (RQP,RPM,RQM,3,3,3,3,3,3)
      CALL MULT3  (RH(1,1,LRPQ),RQN,RPN,3,3,3,3,3,3)
      CALL SKEWV3 (DH(1,LDNQ),SNQ,3,3)
      CALL SKEWV3 (DH(1,LDMP),SMP,3,3)
C
      IQ = LOCU(NOBQ) - 1
      IP = LOCU(NOBP) - 1
      DO 25 I=1,3
      WQN(I) = -Y(IQ+I)
   25 WPM(I) =  Y(IP+I)
      IF (LEQ .EQ. 0) GO TO 26
      DO 27 I=1,3
   27 WQN(I) = WQN(I) - SXDQN(I)
   26 IF (LEP .EQ. 0) GO TO 28
      DO 29 I=1,3
   29 WPM(I) = WPM(I) + SXDPM(I)
   28 CALL MULT3  (RPM,WPM,WPQP,3,3,1,3,1,1)
      CALL MULTAD (RPN,WQN,WPQP,3,3,1,3,1,1)
      CALL SKEWV3 (WPQP,WSK,1,3)
      CALL MULT3  (WSK,RH(1,1,LRPQ),DRPQ,3,3,3,3,3,3)
      DO 30 I=1,3
      DO 30 J=1,3
   30 DRQP(I,J) = DRPQ(J,I)
C
      CALL MULT3  (DRPQ,RQN,DDTQ(4,4),3,3,3,3,3,6)
      CALL MULT3  (DRQP,RPM,DRQM,3,3,3,3,3,3)
      CALL MULT3  (PINDT,RQN,DDTQ,3,3,3,3,3,6)
      IF (LEQ .EQ. 0) GO TO 35
      CALL MULT3  (RQN,SXDQN,VEC,3,3,1,3,1,1)
      CALL SKEWV3 (VEC,WSK,1,3)
      CALL MULT3  (WSK,RQN,DRQN,3,3,3,3,3,3)
      CALL MULTAD (RH(1,1,LRPQ),DRQN,BDTQ(4,4),3,3,3,3,3,6)
      CALL MULTAD (PIN(1,1,L),DRQN,BDTQ,3,3,3,3,3,6)
      CALL MULT3  (BDTQ(4,4),SNQ,BDTQ(4,1),3,3,3,6,3,6)
      CALL SKEWV3 (HXDQN,WSK,1,3)
      CALL MULTAD (RPN,WSK,BDTQ(4,1),3,3,3,3,3,6)
      CALL MULT3  (BDTQ,SIGH(1,1,LHSQ),BDTQ(1,7),3,3,LEQ,6,3,6)
      CALL MULT3  (BDTQ(4,4),HATH(1,1,LHSQ),BDTQ(4,7),3,3,LEQ,6,3,6)
      GO TO 40
   35 CALL MULT3  (BDTQ(4,4),SNQ,BDTQ(4,1),3,3,3,6,3,6)
   40 CONTINUE
C
      IF (LEP .EQ. 0) GO TO 50
      CALL MULT3  (RPM,SXDPM,VEC,3,3,1,3,1,1)
      CALL SKEWV3 (VEC,WSK,1,3)
      CALL MULT3  (WSK,RPM,BDTP(4,4),3,3,3,3,3,6)
      CALL MULT3  (PINDT,RQM,BDTP(1,1),3,3,3,3,3,6)
      CALL MULT3  (BDTP(4,4),SMP,BDTP(4,1),3,3,3,6,3,6)
      CALL MULTAD (RQP,BDTP(4,4),DRQM,3,3,3,3,6,3)
      CALL MULTAD (PIN(1,1,L),DRQM,BDTP,3,3,3,3,3,6)
      CALL SKEWV3 (HXDPM,WSK,1,3)
      CALL MULTAD (RPM,WSK,BDTP(4,1),3,3,3,3,3,6)
      DO 55 I=1,3
      IP3 = I + 3
      DO 55 J=1,3
      JP3 = J + 3
      BDTP(I,J) = -BDTP(I,J)
      BDTP(IP3,J) = -BDTP(IP3,J)
   55 BDTP(IP3,JP3) = -BDTP(IP3,JP3)
      CALL MULT3  (BDTP(1,1),SIGH(1,1,LHSP),BDTP(1,7),3,3,LEP,6,3,6)
      CALL MULT3  (BDTP(4,4),HATH(1,1,LHSP),BDTP(4,7),3,3,LEP,6,3,6)
      GO TO 60
   50 CALL MULT3  (PINDT,RQM,BDTP,3,3,3,3,3,6)
      CALL MULTAD (PIN(1,1,L),DRQM,BDTP,3,3,3,3,3,6)
      DO 58 I=1,3
      DO 58 J=1,3
   58 BDTP(I,J) = -BDTP(I,J)
   60 CONTINUE
      IF(.NOT.LEQU) RETURN
      WRITE(NOT,1001) L
 1001 FORMAT (' BDTP ARRAY HINGE',I3)
      L11 = 6 + LEP
      CALL WRITES(BDTP,6,L11,6)
      WRITE(NOT,1002) L
 1002 FORMAT (' BDTQ ARRAY HINGE',I3)
      L11 = 6 + LEQ
      CALL WRITES(BDTQ,6,L11,6)
      RETURN
C
  100 DO 70 I=1,3
   70 WQN(I) = -Y(I)
      CALL SKEWV3 (WQN,WSK,1,3)
      CALL MULT3  (RQL,WSK,BDTQ(4,4),3,3,3,3,3,6)
      B2DT = BETAHD(2,1)
      B3DT = BETAHD(3,1)
      DO 75 I=1,3
      DO 75 J=1,3
   75 BDTQ(I,J) = B2DT*RP2(I,J,1) + B3DT*RP3(I,J,1)
C
```

```
      IF(.NOT.LEQU) RETURN
      WRITE(NOT,1002) L
      L11 = 6
      CALL WRITES(BDTQ,6,L11,6)
      RETURN
      END
```

```
      SUBROUTINE BAKSLV (BW,NLAM,V,D,KBW)
C                                                           DEBUG #  52
C
C        USE DECOMPOSED BMB AND V TO COMPUTE LAGRANGE MULTIPLIERS
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BW(KBW,1),V(1),D(1)
      DATA NOT/6 /
      LOGICAL DEBUG,LEQU
      COMMON /LDEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(52))
C
      DO 25 I=2,NLAM
      IM1 = I - 1
      DO 25 J=1,IM1
   25 V(I) = V(I) - BW(J,I)*V(J)
      DO 27 I=1,NLAM
   27 V(I) = V(I)/D(I)
      NL1 = NLAM - 1
      DO 30 I=1,NL1
      L = NLAM - I
      LP1 = L + 1
      DO 30 J=LP1,NLAM
   30 V(L) = V(L) - BW(L,J)*V(J)
C
      IF(.NOT.LEQU) RETURN
      WRITE(NOT,1000)
 1000 FORMAT(' LAGRANGE MULTIPLIERS FROM BAKSLV ARE ')
      CALL WRITES(V,1,NLAM,1)
      RETURN
      END
```

```
      SUBROUTINE SKEWV3(V,SKV,KV,KSKV)
C                                                           DEBUG #  53
C
C                    0      V3    -V2
C          SKV =   -V3      0      V1
C                   V2    -V1      0
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION V(KV,1),SKV(KSKV,1)
C
      SKV(2,3) = V(1,1)
      SKV(3,1) = V(2,1)                                                 0 5406
      SKV(1,2) = V(3,1)                                                 0 5407
      SKV(3,2) = -SKV(2,3)                                              0 5408
      SKV(1,3) = -SKV(3,1)                                              0 5409
      SKV(2,1) = -SKV(1,2)                                              0 5410
      SKV(1,1) = 0.D 0                                                  0 5411
      SKV(2,2) = 0.D 0                                                  0 5412
      SKV(3,3) = 0.D 0                                                  0 5413
C                                                                       0 5414
      RETURN                                                            0 5415
      END                                                               0 5416



      SUBROUTINE MULT3(A,B,Z,NRA,NRB,NCB,KRA,KRB,KRZ)                   0 5417
C                                                        DEBUG #   54   0 5418
C                                                                       0 5419
C     GENERAL MATRIX PRODUCT OF RECTANGULAR MATRICES                    0 5420
C        A(NRA X NRB) * B(NRB X NCB) = Z(NRA X NCB)                     0 5421
C                                                                       0 5422
      IMPLICIT REAL*8(A-H,O-Z)                                          0 5423
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1),WR(100)                     84 5424
C                                                                       0 5425
      DO 20 I=1,NRA                                                     0 5426
      DO 15 J=1,NRB                                                     0 5427
   15 WR(J) = A(I,J)                                                    0 5428
      DO 20 J=1,NCB                                                     0 5429
      S = 0.D 0                                                         0 5430
      DO 30 K=1,NRB                                                     0 5431
   30 S = S + WR(K)*B(K,J)                                              0 5432
   20 Z(I,J) = S                                                        0 5433
C                                                                       0 5434
      RETURN                                                            0 5435
      END                                                               0 5436



      SUBROUTINE MULTAD (A,B,Z,NRA,NRB,NCB,KRA,KRB,KRZ)                 0 5437
C                                                        DEBUG #   55   0 5438
C        MULTIPLY                                                       0 5439
C                    A(NRA,NRB) * B(NRB,NCB)                            0 5440
C        ADD THE RESULT TO                                              0 5441
C                    Z(NRA,NRB)                                         0 5442
C        RETURN                                                         0 5443
C                    Z = Z + A * B                                      0 5444
      IMPLICIT REAL*8(A-H,O-Z)                                          0 5445
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1),WR(100)                     85 5446
C                                                                       0 5447
      DO 20 I=1,NRA                                                     0 5448
      DO 15 J=1,NRB                                                     0 5449
   15 WR(J) = A(I,J)                                                    0 5450
      DO 20 J=1,NCB                                                     0 5451
      S = 0.D 0                                                         0 5452
      DO 30 K=1,NRB                                                     0 5453
   30 S = S + WR(K)*B(K,J)                                                     0 5454
   20 Z(I,J) = Z(I,J) + S                                                      0 5455
C                                                                              0 5456
      RETURN                                                                   0 5457
      END                                                                      0 5458



      SUBROUTINE MLTSR (A,B,C,LE,LM,L,IV,KMU)                                  0 5459
C                                                             DEBUG #  56      0 5460
C        USED TO PERFORM MATRIX                                                0 5461
C                    C = A * B                                                 0 5462
C        WHERE                                                                 C 5463
C                    C(6,LM) = A(6,LE ) * B(LE,LM)                             0 5464
C        FOR SELECTED ROWS OF C, SELECTION DEFINED BY IV INTEGER ARRAY         0 5465
      IMPLICIT REAL*8(A-H,O-Z)                                                 0 5466
      DIMENSION A(6,1),B(KMU,1),C(6,1),IV(6,1),RW(13)                         86 5467
C                                                                              0 5468
      DO 10 I=1,6                                                              0 5469
      IL = IV(I,L)                                                             0 5470
      IF (IL .EQ. 0) GO TO 10                                                  0 5471
      DO 15 K=1,LE                                                             0 5472
   15 RW(K) = A(I,K)                                                           0 5473
      DO 20 J=1,LM                                                             0 5474
      S = 0.D0                                                                 0 5475
      DO 25 K=1,LE                                                             0 5476
   25 S = S + RW(K)*B(K,J)                                                     0 5477
   20 C(I,J) = S                                                               0 5478
   10 CONTINUE                                                                 0 5479
C                                                                              0 5480
      RETURN                                                                   0 5481
      END                                                                      C 5482



      SUBROUTINE TORQUE (G)                                                    0 5483
C                                                             DEBUG #  57      0 5484
      IMPLICIT REAL*8(A-H,O-Z)                                                 0 5485
      DIMENSION G(1)                                                           0 5486
C                                                                              0 5487
            COMMON /BHDSRD/                                                    0 5488
     *      BH(6,18,11),BS(6,18,15),ROL(3,3, 6),DDL(3, 6)                      2 5489
            COMMON /GGSAVE/                                                    0 5490
     *      GGS(12,9, 6)                                                       3 5491
            COMMON /INTGRL/                                                    0 5492
     *      A4(171, 6),ACOF(9,12, 6),DCOF(6,12, 6),                            5 5493
     *      COF11(12,12, 6),COF22(12,12, 6),COF33(12,12, 6),AK(12,12, 6),      6 5494
     *      COF12(12,12, 6),COF13(12,12, 6),COF23(12,12, 6),AD(12,12, 6),      7 5495
     *      COFXY(12,12, 6),COFXZ(12,12, 6),COFYZ(12,12, 6)                    8 5496
            COMMON /MAXMUM/                                                    0 5497
     *      NEMAX,NHMAX,NSPMAX,NWWMAX,NMWBCD,NMDBDD,KMU,KY,KU                  0 5498
            COMMON /MOMENG/                                                    0 5499
     *      F(113),FMOM(36),HTOT(3),TOTL(3),ENGKE( 6),ENGPE( 6),              11 5500
     *      TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                                  0 5501
```

```
              COMMON /NUMBRS/                                            0 5502
     *      ZRO,ONE,TWO,TRES                                             0 5503
              COMMON /SPECIF/                                            0 5504
     *      BETAH(6, 6),BETAHD(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),   16 5505
     *      DH(3,35),DS(3,30),IMO(3, 5),NMOW(6, 5),IFTSMW(15),          17 5506
     *      NB,NH,NSPT,NGFMO,NDELTA,ITOPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 5507
     *      LOCO(14),LEND(14),ND,NBETA,NLAW,NEQ                         19 5508
              COMMON /VECTOR/                                            0 5509
     *      Y(250),YDT(250)                                             20 5510
C                                                                        0 5511
      DIMENSION CW(12,3),RW(3,12),V1(12),WSK(3,3),ISHFT( 5),            87 5512
     *      TEX(6,15),ISPN(15),V(6),V2(12)                              88 5513
      DATA NOT/6/                                                        0 5514
      LOGICAL DEBUG,LEQU                                                 0 5515
      COMMON /LDEBUG/ DEBUG(120)                                         0 5516
      EQUIVALENCE (LEQU,DEBUG(57))                                       0 5517
      IF(LEQU) WRITE(NOT,1000)                                           0 5518
 1000 FORMAT (' SUBROUTINE TORQUE ')                                     0 5519
C                                                                        0 5520
CCC   SUBROUTINE CONTRL ESTABLISHES THE D/DT(DELTAS)  USER SUPPLIED --   0 5521
      CALL CONTRL                                                        0 5522
C                                                                        0 5523
CCC   SUBROUTINE EXTOR ESTABLISHES ALL EXTERNAL TORQUES, INCLUDING RCS   0 5524
CCCC  CONTROL TORQUES, ETC.,  USER SUPPLIED --                           0 5525
      CALL EXTOR (TEX,ISPN,NTEX)                                         0 5526
      IF (NTEX .EQ. 0) GO TO 5                                           0 5527
C                                                                        0 5528
      DO 65 L=1,NTEX                                                     0 5529
      NSP = ISPN(L)                                                      0 5530
      NBOD = IFTSMW(NSP)                                                 0 5531
      LON = LOCO(NBOD) - 1                                               0 5532
      LEN = IRGFLX(NBOD) + 6                                             0 5533
      DO 66 I=1,6                                                        0 5534
      TQ = TEX(I,L)                                                      0 5535
      IF (TQ .EQ. ZRO) GO TO 66                                          0 5536
      DO 67 J=1,LEN                                                      0 5537
      JL = J + LON                                                       0 5538
   67 G(JL) = G(JL) + TQ*BS(I,J,NSP)                                     0 5539
   66 CONTINUE                                                           0 5540
   65 CONTINUE                                                           0 5541
C                                                                        0 5542
CCC   SUBROUTINE SHAFTT ESTABLISHES SHAFT TORQUE FOR EACH                0 5543
CCCC  MOMENTUM WHEEL (ZEROS IT OUT IF CONSTANT SPEED) USER SUPPLIED --   0 5544
    5 CALL SHAFTT (TSHFT)                                                0 5545
C                                                                        0 5546
CCCC   SETUP HINGE SPRING AND DASHPOT RESTORING TORQUES                  0 5547
CCCC   ALSO ACCOUNT FOR POTENTIAL ENERGY DUE TO HINGE SPRINGS            0 5548
      CALL KHINGE (G)                                                    0 5549
C                                                                        0 5550
      IF(.NOT.LEQU) GO TO 80                                             0 5551
      WRITE (NOT,1001)                                                   0 5552
 1001 FORMAT (' G VECTOR ALL FORCES AND TORQUES EXTERNAL TO BODIES ')    0 5553
      CALL WRITES(G,1,ND,1)                                              0 5554
   80 CONTINUE                                                           0 5555
C                                                                        0 5556
      KM = NMOBOD                                                        0 5557
C                                                                        0 5558
C        CYCLE THROUGH ALL BODIES COMPUTING GYROSCOPIC FORCES AND        0 5559
C        TORQUES ACTING ON THEM                                          0 5560
      DO 90 N=1,NB                                                       0 5561
      LOU = LOCU(N)
      LO = LOCO(N+NB)
      LE = LENG(N+NB)
      LEU = LE + 6
      IF (LE .EQ. 0) LEU = 3
      LOU1 = LOU + 1
      LOU2 = LOU + 2
      LOU3 = LOU + 3
      LOU4 = LOU + 4
      LOU5 = LOU + 5
C
C        EVALUATE EQUATION II-58 ADD IT TO THE LOAD VECTOR G
      CALL SKEWV3 (Y(LOU),WSK,1,3)
      CALL MULTAD (WSK,P(LOU),G(LOU),3,3,1,3,1,1)
      CALL MULTAD (WSK,P(LOU3),G(LOU3),3,3,1,3,1,1)
      CALL SKEWV3 (Y(LOU3),WSK,1,3)
      CALL MULTAD (WSK,P(LOU3),G(LOU),3,3,1,3,1,1)
      IF (LE .EQ. 0) GO TO 100
C
C        ADD IN THE MODAL STIFFNESS AND DAMPING CONTRIBUTIONS
C        GMISC USED TO ADD IN MISCELLANIOUS EFFECTS SUCH AS
C        THERMALLY INDUCED ACTION
      CALL GMISC (N,LE,LO,V2)
      CALL MULT3  (AK(1,1,N),V2,V1,LE,LE,1,KM,1,1)
      CALL MULTAD (AD(1,1,N),YDT(LO),V1,LE,LE,1,KM,1,1)
      DO 10 J=1,LE
      I = LOU5 + J
   10 G(I) = G(I) - V1(J)
C
C        GGS COMPUTED IN MGEN USED HERE IN THE EVALUATION OF
C        EQUATION II-91,92,93 AS WELL AS IN THE GRAVITY GRADIENT STUFF
      DO 15 J=1,LE
      CW(J,1) = -TWO*Y(LOU )*(BCOF(1,J,N) + GGS(J,1,N))
     *    + Y(LOU1)*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))
     *    + Y(LOU2)*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))
     *  -Y(LOU3)*ACOF(1,J,N) - Y(LOU4)*ACOF(2,J,N) - Y(LOU5)*ACOF(3,J,N)
      CW(J,2) = -TWO*Y(LOU1)*(BCOF(2,J,N) + GGS(J,2,N))
     *    + Y(LOU2)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))
     *    + Y(LOU )*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))
     *  -Y(LOU3)*ACOF(4,J,N) - Y(LOU4)*ACOF(5,J,N) - Y(LOU5)*ACOF(6,J,N)
      CW(J,3) = -TWO*Y(LOU2)*(BCOF(3,J,N) + GGS(J,3,N))
     *    + Y(LOU )*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))
     *    + Y(LOU1)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))
     *  -Y(LOU3)*ACOF(7,J,N) - Y(LOU4)*ACOF(8,J,N) - Y(LOU5)*ACOF(9,J,N)
   15 CONTINUE
      CALL MULTAD (YDT(LO),CW,G(LOU),1,LE,3,1,KM,1)
      CALL MULT3  (COFXY(1,1,N),YDT(LO),CW(1,1),LE,LE,1,KM,1,KM)
      CALL MULT3  (COFXZ(1,1,N),YDT(LO),CW(1,2),LE,LE,1,KM,1,KM)
      CALL MULT3  (COFYZ(1,1,N),YDT(LO),CW(1,3),LE,LE,1,KM,1,KM)
      CALL MULT3  (YDT(LO),CW,V,1,LE,3,1,KM,1)
      G(LOU ) = G(LOU ) - V(3)
      G(LOU1) = G(LOU1) - V(2)
      G(LOU2) = G(LOU2) - V(1)
C
C        COMPUTE EQUATION II-94,95,96
      DO 16 J=1,LE
      CW(J,1) = -Y(LOU )*ACOF(1,J,N) - Y(LOU1)*ACOF(4,J,N)
     *     -Y(LOU2)*ACOF(7,J,N)
      CW(J,2) = -Y(LOU )*ACOF(2,J,N) - Y(LOU1)*ACOF(5,J,N)
     *     -Y(LOU2)*ACOF(8,J,N)
```

```
      CW(J,3) = -Y(LQQ )*ACOF(3,J,N) - Y(LQQ1)*ACOF(6,J,N)
     *     -Y(LQQ2)*ACOF(9,J,N)
   10 CONTINUE
      CALL MULTAD (YDT(LO),CW,G(LQQ3),1,LE,3,1,KM,1)
C
C
C         COMPUTE EQUATION II-97
      DO 20 J=1,LE
      I = LQQ5 + J
      G(I) = G(I) + (Y(LQQ )**2)*(BCOF(1,J,N) + GGS(J,1,N))
     *              + (Y(LQQ1)**2)*(BCOF(2,J,N) + GGS(J,2,N))
     *              + (Y(LQQ2)**2)*(BCOF(3,J,N) + GGS(J,3,N))
     *  - Y(LQQ )*Y(LQQ1)*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))
     *  - Y(LQQ )*Y(LQQ2)*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))
     *  - Y(LQQ1)*Y(LQQ2)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))
     *   + Y(LQQ )*(Y(LQQ3)*ACOF(1,J,N) + Y(LQQ4)*ACOF(2,J,N)
     *     + Y(LQQ5)*ACOF(3,J,N))
     *   + Y(LQQ1)*(Y(LQQ3)*ACOF(4,J,N) + Y(LQQ4)*ACOF(5,J,N)
     *     + Y(LQQ5)*ACOF(6,J,N))
     *   + Y(LQQ2)*(Y(LQQ3)*ACOF(7,J,N) + Y(LQQ4)*ACOF(8,J,N)
     *     + Y(LQQ5)*ACOF(9,J,N))
   20 CONTINUE
C
      CALL MULT3  (COFXY(1,1,N),YDT(LC),CW(1,1),LE,LE,1,KM,1,KM)
      CALL MULT3  (COFXZ(1,1,N),YDT(LC),CW(1,2),LE,LE,1,KM,1,KM)
      CALL MULT3  (COFYZ(1,1,N),YDT(LC),CW(1,3),LE,LE,1,KM,1,KM)
      CALL MULT3  (YDT(LO),COFXY(1,1,N),RW(1,1),1,LE,LE,1,KM,3)
      CALL MULT3  (YDT(LO),COFXZ(1,1,N),RW(2,1),1,LE,LE,1,KM,3)
      CALL MULT3  (YDT(LO),COFYZ(1,1,N),RW(3,1),1,LE,LE,1,KM,3)
      DO 30 J=1,LE
      I = LQQ5 + J
      G(I) = G(I) + Y(LQQ )*(CW(J,3) - RW(3,J))
     *              + Y(LQQ1)*(CW(J,2) - RW(2,J))
     *              + Y(LQQ2)*(CW(J,1) - RW(1,J))
   30 CONTINUE
C
  100 NMON = NMOW(1,N)
C
C         GET CONTRIBUTATIONS DUE TO WHEELS EQUATION II-109
      IF (NMON .EQ. 0) GO TO 50
      IC = 0
      DO 35 I=1,NMON
      IP2 = I + 2
      NW = NMOW(IP2,N)
C         CHECK TO SEE IF WHEEL NW IS ACTIVE
      IF (IMO(3,NW) .EQ. 0) GO TO 37
      IC = IC + 1
      LMO = LQQ5 + LE + IC
      TDTJ = Y(LMO)*AMO(2,NW)
      GO TO 38
   37 TDTJ = AMO(1,NW)*AMO(2,NW)
   38 NPTS = IMO(1,NW)
      NAX  = IMO(2,NW)
      CALL MULT3  (BS(1,1,NPTS),Y(LQQ),V(4),3,LE3,1,6,1,1)
      GO TO (41,42,43), NAX
   41 V(1) = ZRO
      V(2) = -V(6)*TDTJ
      V(3) =  V(5)*TDTJ
      GO TO 40
   42 V(1) =  V(6)*TDTJ
```

```
      V(2) = ZRO                                                        0 5682
      V(3) = -V(4)*TDTJ                                                 0 5683
      GO TO 40                                                          0 5684
   45 V(1) = -V(5)*TDTJ                                                 0 5685
      V(2) =  V(4)*TDTJ                                                 0 5686
      V(3) = ZRO                                                        0 5687
   40 CALL MULTAD (V.BS(1.1.NPTS).G(LCU).1.3.LEU.1.5.1)                 0 5688
      IF (IMO(3.NW) .EQ. 0) GO TO 35                                    0 5689
      G(LMO) = ISHFT(NW)                                                0 5690
      IF (LE .EQ. 0) GO TO 35                                           0 5691
      CALL MULT3  (BS(1.7.NPTS).YDT(LC).V.3.LE.1.6.1.1)                 0 5692
      CALL SKEWV3 (V.WSK.1.3)                                           0 5693
      CALL MULT3  (WSK.V(4).V.3.3.1.3.1.1)                              0 5694
      G(LMO) = G(LMO) - AMO(2.NW)*V(NAX)                                0 5695
   35 CONTINUE                                                          0 5696
C                                                                       0 5697
   50 CONTINUE                                                          0 5698
C                                                                       C 5699
CCC   SUBROUTINE EQADD ESTABLISHES ADDITIONAL CONTROL BLOCK EQUATIONS   0 5700
CCCC  TO SET UP SIMILARITY TRANSFORMATION. USED ONLY FOR LINEARIZATION  0 5701
CCCC  AND STABILITY PACKAGE.  USER SUPPLIED --                          0 5702
      CALL EQADD                                                        0 5703
C                                                                       C 5704
      IF(.NOT.LEQU) RETURN                                              0 5705
      WRITE (NCT.1002)                                                  0 5706
 1002 FORMAT (' LOAD VECTOR G WITH GYROSCOPIC EFFECTS ADDED ON' )       0 5707
      CALL WRITES(G.1.NO.1)                                             0 5708
      RETURN                                                            0 5709
      END                                                               0 5710
```

```
      SUBROUTINE ENGMOM                                                 0 5711
C                                                         DEBUG #  58   0 5712
      IMPLICIT REAL*8(A-H.O-Z)                                          C 5713
C                                                                       0 5714
C        FOR THEORY OF EQUATIONS PROGRAMMED SEE VOL 1 APPENDIX D        C 5715
            COMMON /SHBSRD/                                             0 5716
     *      BH(6.18.11).BS(6.16.15).ROL(3.3. 6).DOL(3. 6)               2 5717
            COMMON /INTGRL/                                             0 5718
     *      AI(171, 6).ACCF(9.12. 6).BCCF(6.12. 6).                     5 5719
     *      CCF11(12.12. 6).CCF22(12.12. 6).CCF33(12.12. 6).AK(12.12. 6).  6 5720
     *      CCF12(12.12. 6).CCF13(12.12. 6).CCF23(12.12. 6).AD(12.12. 6).  7 5721
     *      CCFXY(12.12. 6).CCFXZ(12.12. 6).CCFYZ(12.12. 6)            8 5722
            COMMON /MAXMUM/                                             0 5723
     *      NJMAX.NMMAX.NSPMAX.NMWMAX.NMW200.NMJBOD.KMJ.KY.KU           0 5724
            COMMON /MOMENG/                                             0 5725
     *      P(113).PNG4(30).HTOT(3).TOTL(3).ENGKE( 6).ENGPE( 6).       11 5726
     *      TOTKE. TOTPE. TOTENG. AHTOT.ATOTL                           0 5727
            COMMON /NUMBRS/                                             0 5728
     *      ZRO.ONE.TWO.TRES                                            0 5729
            COMMON /SPECIF/                                             0 5730
     *      BETAH(6. 6).BETAHO(6. 6).AMO(2. 5).RH(3.3.30).RS(3.3.30).  16 5731
     *      DH(3.30).DS(3.30).IMO(3. 5).NMOW(6. 5).IFTSAW(15).         17 5732
     *      NJ.NM.NSPT.NSWMO.NDELTA.ITOPOL(2. 6).IRGFLX( 6).IMDATA(7. 6).  18 5733
     *      LOCO(14).LENO(14).NO.NDELTA.NLAM.NEQ                       19 5734
            COMMON /VECTOR/                                             0 5735
     *     Y(250),YDT(250)
C
      DIMENSION WV(12)
      DIMENSION VW(3)
C
      KM = NMCBOD
      DO 5 I=1,3
      HTOT(I) = ZRO
    5 TOTL(I) = ZRO
      DO 10 N=1,NB
      LA = 6*N - 5
      LL = LA + 3
      LOA = LOCO(N)
      LOL = LOCO(N) + 3
      CALL MULT3  (ROL(1,1,N),P(LOA),PMOM(LA),3,3,1,3,1,1)
      CALL MULT3  (ROL(1,1,N),P(LOL),PMOM(LL),3,3,1,3,1,1)
      PMOM(LA  ) = PMOM(LA  )+DOL(2,N)*PMOM(LL+2)-DOL(3,N)*PMOM(LL+1)
      PMOM(LA+1) = PMOM(LA+1)+DOL(3,N)*PMOM(LL  )-DOL(1,N)*PMOM(LL+2)
      PMOM(LA+2) = PMOM(LA+2)+DOL(1,N)*PMOM(LL+1)-DOL(2,N)*PMOM(LL  )
CCC   STATEMENTS THRU 40 TO ACCOUNT FOR ANGULAR MOMENTUM DUE TO
CCC   CONSTANT SPEED MOMENTUM WHEELS.
      NM = NMCW(1,N)
      IF (NM .EQ. 0) GO TO 40
      DO 30 I=1,NM
      NW = NMCW(2+I,N)
      IF (IMO(3,NW) .NE. 0) GO TO 30
      NA = IMO(2,NW)
      NS = IMO(1,NW)
      PH = AMO(1,NW)*AMO(2,NW)
      DO 35 J=1,3
   35 VW(J) = PH*BS(NA,J,NS)
      CALL MULTAD (ROL(1,1,N),VW,PMOM(LA),3,3,1,3,1,1)
   30 CONTINUE
   40 CONTINUE
      DO 15 I=1,3
      I1 = LA - 1 + I
      I2 = LL - 1 + I
      HTOT(I) = HTOT(I) + PMOM(I1)
   15 TOTL(I) = TOTL(I) + PMOM(I2)
   10 CONTINUE
C
      TOTKE = ZRO
      DO 20 N=1,NB
      LOO = LOCO(N)
      LO  = LOCO(N+NB)
      ENGPE(N) = ZRO
      LE   = LENO(N+NB)
      LEO = LENO(N)
      IF (LE .EQ. 0) GO TO 22
      CALL MULT3  (AK(1,1,N),Y(LO),WV,LE,LE,1,KM,1,1)
      CALL MULT3  (Y(LO),WV,ENGPE(N),1,LE,1,1,1,1)
      ENGPE(N) = ENGPE(N)/TWO
   22 CALL MULT3  (Y(LOO),P(LOO),ENGKE(N),1,LEO,1,1,1,1)
      ENGKE(N) = ENGKE(N)/TWO
      TOTKE = TOTKE + ENGKE(N)
   20 TOTPE = TOTPE + ENGPE(N)
      TOTENG = TOTKE + TOTPE
      ATOTL = DSQRT(TOTL(1)**2 + TOTL(2)**2 + TOTL(3)**2)
      AHTOT = DSQRT(HTOT(1)**2 + HTOT(2)**2 + HTOT(3)**2)
C
      RETURN
      END
      SUBROUTINE RKADAM(NEQ)                                                0 5798
C                                                          DEBUG #  59      0 5799
      IMPLICIT REAL*8(A-H,O-Z)                                              0 5800
C                                                                           0 5801
            COMMON /DIEFLG/                                                 0 5802
     *      DIE                                                             0 5803
            COMMON /PRWORK/                                                 0 5804
     *      PR(250,5)                                                      14 5805
            COMMON /QPRKTA/                                                 0 5806
     *      QRK(250),PRK(4), NT                                            15 5807
            COMMON /TIMESS/                                                 0 5808
     *      STARTT,DELTAT,T,ENDT,TMST                                       0 5809
            COMMON /VECTOR/                                                 0 5810
     *      Y(250),YDT(250)                                                20 5811
            COMMON /VINDEP/                                                 0 5812
     *      I,DEP(250)                                                     21 5813
C                                                                           0 5814
      DATA EPS1, EPS2  /  1.D-5,  1.D-2   /                                 0 5815
      DATA Q,T,MAXIT / 0, 10/                                               0 5816
C                                                                           0 5817
C  *****     ITYPE .EQ. 1        RUNGE-KUTTA INTEGRATION                    0 5818
C  *****     ITYPE .EQ. 2        ADAMS PREDICTOR/CORRECTOR INTEGRATION      0 5819
C                                                                           0 5820
C  SUBROUTINE TO INTEGRATE DIFFERENTIAL EQUATIONS (FIRST ORDER)             0 5821
C  IN THE TIME DOMAIN. USES RUNGE-KUTTA-GILL TO START THE ADAMS PREDICT     0 5822
C       CORRECTOR. MAY USE RUNGE-KUTTA ONLY. ON OPTION.                     0 5823
C    CODED BY CARL BODLEY  1971                                             0 5824
C   MODIFICATION TO CORRECTOR LOOP TO ACCOUNT FOR IMPULSIVE CHANGE TO       0 5825
C     STATE VECTOR (U)TWHICH OCCURS IN SUB. YDOT. MADE BY CARL BODLEY       0 5826
C     APR, 1974                                                             0 5827
C                                                                           0 5828
      DATA ITYPE /   1   /                                                  0 5829
C                                                                           0 5830
      GO TO (10,20) , ITYPE                                                 0 5831
C                                                                           0 5832
   20 IF (NT .GT. 0) GO TO 201                                              0 5833
      DNM = DELTAT/24.D 0                                                   0 5834
      TR1 =  DNM*55.D 0                                                     0 5835
      TR2 = -DNM*59.D 0                                                     0 5836
      TR3 =  DNM*37.D 0                                                     0 5837
      TR4 = -DNM* 9.D 0                                                     0 5838
      TR5 =  DNM* 9.D 0                                                     0 5839
      TR6 =  DNM*19.D 0                                                     0 5840
      TR7 = -DNM* 5.D 0                                                     0 5841
      TR8 =  DNM* 1.D 0                                                     0 5842
C                                                                           0 5843
  201 IF (NT .GT. 3) GO TO 200                                              0 5844
      NL =    NT + 1                                                        0 5845
      DO 205 I=1,NEQ                                                        0 5846
      PR(I,NL) = YDT(I)                                                     0 5847
  205 PR(I, 5) = Y  (I)                                                     0 5848
      IF (NT .EQ. 3) GO TO 200                                              0 5849
```

```
C
   10 DO 120 J = 1,4
      JIL = J
      DO 110 I=1,NEQ
      Z = YDT(I)*DELTAT
      GO TO (103,101,101,105), JIL
  101 R = PRK(JIL)*(Z - QRK(I))
      GO TO 107
  103 R = PRK(JIL)*Z - QRK(I)
      GO TO 107
  105 R = (Z - 2.D 0*QRK(I))/6.D 0
  107 Y(I) = Y(I) + R
      QRK(I) = QRK(I) + 3.D 0*R - PRK(JIL)*Z
  110 CONTINUE
      IF (JIL .EQ. 1 .OR. JIL .EQ. 3) T = T + DELTAT/2.D 0
  120 CALL YDOT
      GO TO 300
C
  200 DO 204 I=1,NEQ
  204 Y(I) = PR(I,5)+TR1*PR(I,4)+TR2*PR(I,3)+TR3*PR(I,2)+TR4*PR(I,1)
      T = T + DELTAT
      ITER = 0
  207 CALL YDOT
C
      G = 0.D 0
      DO 203 I=1,NEQ
      IF (INDEP(I) .EQ. 0) GO TO 203
      YC     = PR(I,5)+TR5*YDT(I) +TR6*PR(I,4)+TR7*PR(I,3)+TR8*PR(I,2)
      DN = DABS(Y(I))
      DN1 = DABS(YC)
      IF (DN1 .GT. DN) DN = DN1
      IF (DN .LT. EPS1) GO TO 203
      G1 = DABS(YC - Y(I))/DN
      IF (G1 .GT. G) G = G1
      Y(I) = YC
  203 CONTINUE
      ITER = ITER + 1
      IF (G .LE. EPS2) GO TO 30
      IF (ITER .EQ. MAXIT) GO TO 999
      GO TO 207
C
   30 DO 210 I=1,NEQ
      PR(I,1) = PR(I,2)
      PR(I,2) = PR(I,3)
      PR(I,3) = PR(I,4)
      PR(I,4) = YDT(I)
  210 PR(I,5) = Y(I)
C
  300 NT = NT + 1
      ANT = NT
      TMST = ANT*DELTAT
      T = STARTT + TMST
C
      RETURN
  999 WRITE (NOT,1001) MAXIT
 1001 FORMAT (1H1,31HCORRECTOR FAILS TO CONVERGE IN I3,
     *     28HITERATIONS, PROGRAM STOPPED.)
      STOP
      END
      SUBROUTINE PRNTOU                                                     0 5909
C                                                           DEBUG #  60     0 5910
      IMPLICIT REAL*8(A-H,O-Z)                                              0 5911
      REAL*4 CTNEW,TNEW,CTOLD,TOLD,CPSTEP,CPSEC                             0 5912
C                                                                           0 5913
              COMMON /AMOBW /                                               0 5914
     *        AMO(22,22, 6),BW(30,113)                                      1 5915
              COMMON /BHBSRJ/                                               0 5916
     *        BH(6,16,11),BS(6,16,15),RJL(3,3, 6),JJL(3, 6)                 2 5917
              COMMON /MAXNUM/                                               0 5918
     *        NOMAX,NHMAX,NCPMAX,NMWMAX,NMWBOD,NMDJJJ,K4J,KY,KU             0 5919
              COMMON /LAMBDA/                                               0 5920
     *        ALAM(30)                                                     10 5921
              COMMON /MISCNO/                                               0 5922
     *        NOPRNT,NOPLOT                                                 0 5923
              COMMON /MOMENS/                                               0 5924
     *        P(15),PROM(30),HTOT(3),TOTL(3),ENGKE( 6),ENGPE( 6),          11 5925
     *        TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                             0 5926
              COMMON /SPECIF/                                               0 5927
     *        BETAH(6, 6),BETAHD(6, 6),AMC(2, 5),RH(3,3,30),RS(3,3,30),    16 5928
     *        CH(3,35),DS(3,30),IMO(3, 5),NMOW(6, 6),IFTSAW(15),           17 5929
     *        NB,NH,NSPT,NOFMO,NDELTA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 5930
     *        LOCC(14),LENO(14),NU,NBETA,NLAM,NEQ                          19 5931
              COMMON /TIMESS/                                               0 5932
     *        STARTT,DELTAT,T,ENDT,TMST                                     0 5933
              COMMON /VECTOR/                                               0 5934
     *        Y(250),YDT(250)                                              20 5935
C                                                                           0 5936
      DIMENSION V1(3), V2(3)                                                0 5937
      DIMENSION T1(3,3), T2(3,3), TRN(3,3)                                  0 5938
      DIMENSION VCM(3, 6),TIN(3,3, 6),SYSCM(3),SYSIN(3,3),AMAS( 6)         93 5939
      DATA NOT,IIST / 6, 0 /                                                0 5940
C                                                                           0 5941
 1000 FORMAT (//10X,24HAT SIMULATION TIME, T = , 1PD10.4,32(2H* ))          0 5942
 1001 FORMAT ( 3X,21HTHE STATE VECTOR Y = )                                 0 5943
 1002 FORMAT ( 3X,39HTHE STATE VECTOR TIME DERIVATIVE YDT = )               0 5944
 1003 FORMAT ( 3X,53HTHE BETAS (EULER ANGLES, POSITION COORDINATES) ARE)    0 5945
 1004 FORMAT ( 3X,29HTHE BETA TIME DERIVATIVES ARE)                         0 5946
 1005 FORMAT ( 3X,41HTHE DELTAS (CONTROL SYSTEM VARIABLES) ARE)             0 5947
 1006 FORMAT ( 3X,30HTHE DELTA TIME DERIVATIVES ARE)                        0 5948
 1007 FORMAT ( 3X, 9HFOR BODY ,I2,3X,18HTHE VELOCITIES ARE)                 0 5949
 1017 FORMAT ( 3X, 9HFOR BODY ,I2,3X,29HTHE CORRESPONDING MOMENTA ARE)      0 5950
 1027 FORMAT ( 3X, 9HFOR BODY ,I2,3X,25HITS CONTRIBUTION TO TOTAL,          0 5951
     *     23H ANGULAR AND LINEAR MOMENTUM IS)                              0 5952
 1008 FORMAT ( 3X,48HITS CONTRIBUTION TO TOTAL KINETIC AND POTENTIAL ,      0 5953
     *   12H ENERGIES IS ,3X,1P2D15.6)                                      0 5954
 1009 FORMAT ( 3X, 9HFOR BODY ,I2,3X,27HTHE ELASTIC DEFLECTIONS ARE)        0 5955
 1010 FORMAT ( 3X,50HTHE INTERCONNECTION CONSTRAINT FORCES(LAMBDAS) ARE)    0 5956
 1011 FORMAT ( 3X,30HTHE TOTAL ANGULAR MOMENTUM VECTOR IS)                  0 5957
 1012 FORMAT ( 3X,35HTHE TOTAL LINEAR MOMENTUM VECTOR IS)                   0 5958
 1013 FORMAT (/3X,29HTHE TOTAL ANGULAR MOMENTUM = ,1PD15.8,                 0 5959
     *         / 3X,29HTHE TOTAL LINEAR MOMENTUM  = ,1PD15.8,               0 5960
     *         / 3X,29HTHE TOTAL KINETIC ENERGY   = ,1PD15.8,               0 5961
     *         / 3X,29HTHE TOTAL POTENTIAL ENERGY = ,1PD15.8,               0 5962
     *         / 3X,29HTHE TOTAL ENERGY (T + V)   = ,1PD15.8)               0 5963
 1014 FORMAT (//35X,33HCPU TIME/STEP  CPU TIME/REAL TIME, /38X,1PE10.4,     0 5964
     *             9X,1PE10.4)                                              0 5965
C                                                                           0 5966
      IF (IIST .EQ. 1) GO TO 5                                              0 5967
      CTNEW = 0.                                                            0 5968
      INEW = STARTT
    5 CALL PAGEHD
      WRITE (NCT,1000) T
      WRITE (NCT,1001)
      CALL WRITES (Y,1,NEQ,1)
C
      WRITE (NCT,1000) T
      WRITE (NCT,1002)
      CALL WRITES (YDT,1,NEQ,1)
C
      WRITE (NCT,1000) T
      WRITE (NCT,1003)
      CALL WRITES (BETAH,6,NH,6)
C
      WRITE (NCT,1000) T
      WRITE (NCT,1004)
      CALL WRITES (BETAHD,6,NH,6)
C
      IF (NDELTA .EQ. 0) GO TO 10
      WRITE (NCT,1000) T
      WRITE (NCT,1005)
      LO = LOCO(2*NB + 2)
      CALL WRITES (Y(LO),1,NDELTA,1)
      WRITE (NCT,1000) T
      WRITE (NCT,1006)
      CALL WRITES (YDT(LO),1,NDELTA,1)
C
   10 DO 20 N=1,NB
      WRITE (NCT,1000) T
      WRITE (NCT,1007) N
      LO = LOCO(N)
      LE = LENG(N)
      CALL WRITES (Y(LO),1,LE,1)
      WRITE (NCT,1017) N
      CALL WRITES (P(LO),1,LE,1)
      LOPM = 6*(N-1) + 1
      WRITE (NCT,1027) N
      CALL WRITES (PMOM(LOPM),1,6,1)
      WRITE (NCT,1008) ENGKE(N),ENGPE(N)
      LE = LENG(N+NB)
      IF (LE .EQ. 0) GO TO 31
      LO = LOCO(N+NB)
      WRITE (NCT,1009) N
      CALL WRITES (Y(LO),1,LE,1)
C
C     BODY N TO BODY 1 TRANSFORMATION MATRIX
   31 DO 21 I=1,3
      DO 21 J=1,3
      TRN(I,J) = 0.0D0
      DO 21 K=1,3
      TRN(I,J) = TRN(I,J) + ROL(K,I,1)*ROL(K,J,N)
   21 CONTINUE
      WRITE (NCT,1020) N
 1020 FORMAT ( 3X, 5HFOR BODY ,I2,3X,46HTRANSFORMATION MATRIX TO BODY 1
     *COORDINATES IS)
      CALL WRITES (TRN,3,3,3)
C
C     LOCATION BODY N CM WRT BODY 1 REFERENCE
      LE = LENG(N)
      CALL INVINP(AMU(1,1,N),AMU(1,1,N),LE,KMU)
      XMAS(N) = AMU(4,4,N)
      VI(1) = AMU(3,5,N)/XMAS(N)
      VI(2) = AMU(1,6,N)/XMAS(N)
      VI(3) = AMU(2,4,N)/XMAS(N)
      DO 22 I=1,3
      VCM(I,N) = 0.0D0
      DO 22 K=1,3
      VCM(I,N) = VCM(I,N) + TRN(I,K)*VI(K)
     *                       + RCL(K,I,1)*(DJL(K,N)-DJL(K,1))
   22 CONTINUE
      WRITE (NOT,1015) N
 1015 FORMAT ( 5X, 9HFOR BODY ,I2,3X,57HCM LOCATION WRT BODY 1 REFERENCE
     * POINT AND COORDINATES IS)
      CALL WRITES (VCM(1,N),1,3,1)
C
C     INERTIA TENSOR OF BODY N ABOUT ITS CM WRT BODY N REFERENCE FRAME
      DO 23 I=1,3
      DO 23 J=1,3
      TI(I,J) = AMU(I,J,N)
      DO 23 K=1,3
      TI(I,J) = TI(I,J) - AMU(I,K+3,N)*AMU(K+3,J,N)/XMAS(N)
   23 CONTINUE
C
C     INERTIA TENSOR OF BODY N ABOUT ITS CM WRT BODY 1 REFERENCE
      DO 24 I=1,3
      DO 24 J=1,3
      TIN(I,J,N) = 0.0D0
      DO 24 K=1,3
      DO 24 L=1,3
      TIN(I,J,N) = TIN(I,J,N) + TRN(I,K)*TI(K,L)*TRN(J,L)
   24 CONTINUE
      WRITE (NOT,1016) N
 1016 FORMAT ( 5X, 9HFOR BODY ,I2,3X,71HINERTIA TENSOR RELATIVE TO ITS C
     *ENTER OF MASS WRT BODY 1 COORDINATES IS)
      CALL WRITES (TIN(1,1,N),3,3,3)
      CALL INVINP(AMU(1,1,N),AMU(1,1,N),LE,KMU)
C
   20 CONTINUE
C
      IF (NLAM .EQ. 0) GO TO 50
      WRITE (NOT,1000) T
      WRITE (NOT,1010)
      CALL WRITES (ALAM,1,NLAM,1)
C
   50 WRITE (NOT,1000) T
C
C     COMPOSITE SYSTEM CENTER OF MASS AND INERTIA TENSOR RELATIVE TO IT
C       IN BODY 1 COORDINATES
      XMASS = 0.D0
      DO 25 I=1,3
      SYSCM(I) = 0.0D0
      DO 25 J=1,3
      SYSIN(I,J) = 0.0D0
   25 CONTINUE
C
      DO 26 N=1,NB
      XMASS = XMASS + XMAS(N)
      DO 26 I=1,3
      SYSCM(I) = SYSCM(I) + VCM(I,N)*XMAS(N)
      DO 26 J=1,3
      SYSIN(I,J) = SYSIN(I,J) + TIN(I,J,N)
   26 CONTINUE
      DO 27 I=1,3
   27 SYSCM(I) = SYSCM(I)/XMASS
C
      DO 28 N=1,NB
      DO 29 I=1,3
   29 V1(I) = VCM(I,N) - SYSCM(I)
      CALL SKEWV3(V1,T1,3,3)
      DO 30 I=1,3
      DO 30 J=1,3
      DO 30 K=1,3
      SYSIN(I,J) = SYSIN(I,J) + XMAS(N)*T1(I,K)*T1(J,K)
   30 CONTINUE
   28 CONTINUE
C
      WRITE (NOT,1018)
 1018 FORMAT ( 3X,79HTHE SYSTEM CENTER OF MASS RELATIVE TO BODY 1 REFERE
     *NCE POINT AND COORDINATES IS)
      CALL WRITES(SYSCM,1,3,1)
      WRITE (NOT,1019)
 1019 FORMAT ( 3X,84HTHE SYSTEM INERTIA TENSOR RELATIVE TO SYSTEM CENTER
     * OF MASS IN BODY 1 COORDINATES IS)
      CALL WRITES(SYSIN,3,3,3)
C
      WRITE (NOT,1011)
      CALL WRITES (HTOT,1,3,1)
      WRITE (NOT,1012)
      CALL WRITES (TOTL,1,3,1)
C
      WRITE (NOT,1013) AHTOT,ATOTL,TOTKE,TOTPE,TOTENG
C
      IF (IIST .EQ. 1) GO TO 100
      IIST = 1
      RETURN
  100 TOLD = TNEW
      CTOLD = CTNEW
      TNEW = T
      CALL SECOND (CTNEW)
      CPSTEP = CTNEW - CTOLD
      CPSEC = CPSTEP/(TNEW-TOLD)
      CPSTEP = CPSTEP/FLOAT(NOPRNT)
      WRITE (NOT,1014) CPSTEP,CPSEC
C
      RETURN
      END




      SUBROUTINE PLOTWR
C                                                      DEBUG #  61
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE TO WRITE PLOT OUTPUT TAPE
C
              COMMON /LAMBDA/
     *        ALAM(36)
```

```
          COMMON /MAXMUM/                                                  0 6143
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBCD,NMDBUD,KMJ,KY,KU              0 6144
          COMMON /MOMENG/                                                  0 6145
     *      P(113),PMOM(30),HTOT(3),TOTL(3),ENGKE( 6),ENGPE( 6),          11 6146
     *      TOTKE, TOTPE, TOTENG, AHTCT,ATCTL                              0 6147
          COMMON /PLTDTA/                                                  0 6148
     *      NRPLOT,NCPLOT                                                  0 6149
          COMMON /SPECIF/                                                  0 6150
     *      BETAH(6, 6),BETAHD(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),     16 6151
     *      DH(3,35),DS(3,30),IMD(3, 5),NMOW(6, 5),IFTSMW(15),            17 6152
     *      NB,NH,NSPT,NGFMD,NDELTA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 6153
     *      LOCL(14),LEND(14),ND,NBETA,NLAM,NEQ                           19 6154
          COMMON /TAPENO/                                                  0 6155
     *      NTAPE1,NTAPE2,NTAPE3                                           0 6156
          COMMON /TIMESS/                                                  0 6157
     *      STARTT,DELTAT,T,ENDT,TMST                                      0 6158
          COMMON /VECTOR/                                                  0 6159
     *      Y(250),YDT(250)                                               20 6160
C                                                                          0 6161
C                                                                          0 6162
      DATA IIST / 0 /                                                      0 6163
      IF (IIST .EQ. 1) GO TO 5                                             0 6164
      REWIND NTAPE3                                                        0 6165
      NRPLOT = 0                                                           0 6166
      JR = 6*NB                                                            0 6167
      IIST = 1                                                             0 6168
      NLAMD = NLAM                                                         0 6169
      IF (NLAM .EQ. 0) NLAMD = 1                                           0 6170
      NCPLOT = 1+2*NEQ+NLAMD+ND+JR+3+3+2*NB+5                              0 6171
    5 NRPLOT = NRPLOT + 1                                                  0 6172
C                                                                          0 6173
      WRITE (NTAPE3)  T                                                    0 6174
     *    ,   (Y(J),J=1,NEQ), (YDT(J),J=1,NEQ)                             0 6175
     *    ,   (ALAM(J),J=1,NLAMD), (P(J),J=1,ND)                           0 6176
     *    ,   (PMOM(J),J=1,JR), (HTOT(J),J=1,3)                            0 6177
     *    ,   (TOTL(J),J=1,3), (ENGKE(J),J=1,NB)                           0 6178
     *    ,   (ENGPE(J),J=1,NB), AHTCT,ATCTL,TOTKE,TOTPE,TOTENG            0 6179
C                                                                          0 6180
      RETURN                                                               0 6181
      END                                                                  0 6182
```

```
      SUBROUTINE DYN330                                                    0 6183
C                                                        DEBUG #  62       0 6184
C                                                                          0 6185
      REAL*8 DUM,ICTITL,TITL1,TITL3,PTITLE                                 0 6186
C                                                                          0 6187
      DIMENSION ZP(1000,16),DUM(1500),JVPL(16)                            90 6188
      DIMENSION ICTITL(10),NCD( 3),PTITLE( 6)                              0 6189
C                                                                          0 6190
          COMMON /PLTDTA/                                                  0 6191
     *      NRPLOT,NCPLOT                                                  0 6192
          COMMON /TAPENO/                                                  0 6193
     *      NTAPE1,NTAPE2,NTAPE3                                           0 6194
C                                                                          0 6195
      DATA NIT,NOT /5,6/                                                   0 6196
      DATA KRPLOT,KCPLOT /1000,16/
C
      READ (NIT,1005) (ICTITL(I),I=1,10)
 1005 FORMAT(10A8)
      CALL PAGEHD
      WRITE (NCT,1001) (ICTITL(I),I=1,10)
 1001 FORMAT (///,30X,31HSUMMARY OF PLOTTING INFORMATION//,10X,10A8)
C
      READ (NIT,1003) NSET
 1003 FORMAT(16I5)
      IF (NSET .EQ. 0) RETURN
      WRITE (NCT,1011) NSET,NRPLOT,NCPLOT,KRPLOT,KCPLOT
 1011 FORMAT(//,10X,10HNSET      =I5,/,
     *             10X,10HNRPLOT    =I5,  10X,10HNCPLOT    =I5,/,
     *             10X,10HKRPLOT    =I5,  10X,10HKCPLOT    =I5 )
C
      IF (NRPLOT .LE. KRPLOT) GO TO 1500
      NRPLOT = KRPLOT
      WRITE (NCT,1009)
 1009 FORMAT (//,10X,46HNRPLOT EXCEEDED KRPLOT AND WAS RESET TO KRPLOT)
 1500 CONTINUE
C
      DO 1000 ISET=1,NSET
      REWIND NTAPE3
      READ (NIT,1003) JPL
      IF (JPL .GT. KCPLOT) GO TO 998
      READ (NIT,1003) (JVPL(J),J=1,JPL)
      WRITE (NCT,1012) ISET,(JVPL(J),J=1,JPL)
 1012 FORMAT(//,10X,7HISET  =  I5,/10X,7HJVPL  =16I5)
C
      DO 2000 II=1,NRPLOT
      READ (NTAPE3) (DUM(I),I=1,NCPLOT)
      DO 2001 J=1,JPL
      JC = JVPL(J)
 2001 ZP(II,J) = DUM(JC)
 2000 CONTINUE
C
   20 READ (NIT,1003)  NCI,(NCD(I),I=1,3),NGRID
      IF (NCI .EQ. 0) GO TO 1000
      IF(NCI .GT. 1) NGRID = 1
      IF(NGRID .EQ. 0) NGRID = 1
      READ (NIT,1004) TITLI,TITLD,(PTITLE(I),I=1, 6)
 1004 FORMAT(2(A8,2X),6A8)
      WRITE(NCT,1006) NCI,(NCD(I),I=1,3),NGRID,
     *                 TITLI,TITLD,(PTITLE(I),I=1,6)
 1006 FORMAT(//,15X,7HNCI    =,I5,5X,7HNCD    =,3I5,5X,7HNGRID =,I5,/,
     *                 15X,A8,5X,A8,5X,6A8)
      CALL PLTCAR(ZP,NCI,NCD,NRPLOT,NGRID,
     *   TITLI,TITLD,PTITLE,ICTITL,KRPLOT)
      GO TO 20
C
 1000 CONTINUE
C
      RETURN
C
  998 WRITE (NCT,1020)
 1020 FORMAT(//,10X,34HERROR IN PLOT INPUT DATA, STOP RUN)
      STOP
C
      END
      SUBROUTINE PLTCAR (DATA,NI,NJ,NR,NG,
C                                                         DEBUG #  63
     *                   TITLI,TITLD,TITLP,TITLM,KR)
C
C     DATA          DATA ARRAY
C     NI            LOCATION OF INDEPENDENT VARIABLE (X)
C     NJ            LOCATION OF DEPENDENT VARIABLES  (Y(I),I=1,3)
C     NR            NO OF VALUES TO PLOT
C     NG            NO OF GRIDS
C     TITLI         TITLE FOR INDEPENDENT VARIABLE AXIS
C     TITLD         TITLE FOR DEPENDENT VARIABLE AXIS
C     TITLP         PLOT TITLE - UNIQUE TO FRAME
C     TITLM         PLOT TITLE - ALL FRAMES
C     KR            ROW DIMENSION OF DATA ARRAY IN CALLING PROGRAM
C
      REAL*4 MINI,MAXI,MIND,MAXD
      REAL*8 TITLP,TITLM,TITLI,TITLD,TITL(5),IRUNNO,IDATE
      REAL*8 UNAME,TITLE1,TITLE2
C
      COMMON /ZCMODE/ AMODE(200)
      COMMON /ZSTART/ IRUNNO,IDATE,NPAGE
      COMMON /ZSTRT1/ UNAME(3),TITLE1(12),TITLE2(12)
C
      DIMENSION DATA(KR,1),TITLP(1),TITLM(1),NJ(1)
      DIMENSION VI(1000),VJ(1000)
C
      DATA NKP /12/
C
C     FORM TITL
      TITL(1) = IRUNNO
      TITL(2) = IDATE
      TITL(3) = UNAME(1)
      TITL(4) = UNAME(2)
      TITL(5) = UNAME(3)
C
C     NO OF PLOTS
      NPLOTS = 0
      DO 5 I=1,3
      IF(NJ(I) .NE. 0) NPLOTS = NPLOTS+1
    5 CONTINUE
C
C     FIND MAX/MIN OF DEPENDENT VARIABLES
      J = NJ(1)
      MAXD = DATA(1,J)
      MIND = DATA(1,J)
      DO 10 L=1,NPLOTS
      J = NJ(L)
      DO 10 I=1,NR
      IF(DATA(I,J) .GT. MAXD) MAXD = DATA(I,J)
      IF(DATA(I,J) .LT. MIND) MIND = DATA(I,J)
   10 CONTINUE
      IF(MAXD .EQ. MIND) MAXD = MIND + 10.0
C
C     GRID LOOP ******************************
C
      NLFT = 1
      NDIV = NR/NG
C
      DO 45 II=1,NG
      IF(II .GT.  1) NLFT=NRGT
      NPCT = II*NDIV
      IF(II .EQ. NG) NRGT=NR
      NP   = NRGT - NLFT + 1
C
C     FIND MAX/MIN OF INDEPENDENT VARIABLE
      MAXI = DATA(NLFT,NI)
      MINI = DATA(NLFT,NI)
      DO 11 I=NLFT,NRGT
      IF(DATA(I,NI) .GT. MAXI) MAXI = DATA(I,NI)
      IF(DATA(I,NI) .LT. MINI) MINI = DATA(I,NI)
   11 CONTINUE
      IF(MAXI .EQ. MINI) MAXI = MINI + 10.0
C
C     INITIALIZE MODE SETS
      CALL RSETMG(AMODE)
C
C     SET OBJECT SPACE
      CALL SETSMG(AMODE,19,1.0)
      CALL SETSMG(AMODE,20,1.0)
      CALL OBJCTG(AMODE,0.0,0.0,1.0,1.0)
C
C     BOX FRAME
      CALL SETSMG(AMODE,14,3.00)
      CALL SETSMG(AMODE,30,4.00)
      CALL LINESG(AMODE,0,0.00,0.00)
      CALL LINESG(AMODE,1,0.00,0.99)
      CALL LINESG(AMODE,1,1.00,0.99)
      CALL LINESG(AMODE,1,1.00,0.00)
      CALL LINESG(AMODE,1,0.00,0.00)
C
C     TITLE FRAME
      CALL SETSMG(AMODE,45,1.25)
      CALL LINESG(AMODE,0,0.050,0.020)
      CALL TEXTG (AMODE,8,TITL(1))
      CALL TEXTG (AMODE,8,TITL(2))
      CALL LINESG(AMODE,0,0.300,0.020)
      CALL TEXTG (AMODE,8,TITL(3))
      CALL TEXTG (AMODE,8,TITL(4))
      CALL TEXTG (AMODE,8,TITL(5))
      CALL LEGNDG(AMODE,0.050,0.06,80,TITLM)
      CALL LEGNDG(AMODE,0.300,0.10,48,TITLP)
C
C     SET OBJECT/SUBJECT SPACE
      CALL OBJCTG(AMODE,0.01,0.05,1.0 ,1.0 )
      CALL SUBJEG(AMODE,MINI,MIND,MAXI,MAXD)
C
C     CARTESIAN GRID AND LABELS
      CALL SETSMG(AMODE,14,0.00)
      CALL SETSMG(AMODE,45,1.00)
      CALL SETLPG(AMODE,1,DI,DD,IITH,IDTH,DLI,DLD,FMTI,FMTD)
      DO 12 I=106,109
   12 AMODE(I) = 0.0
      CALL GRIDG (AMODE,  DI,DD,IITH,IDTH)
      CALL LABELG(AMODE,0,DLI,0, FMTI)
      CALL LABELG(AMODE,1,DLD,0, FMTD)
      CALL TITLEG(AMODE,8,TITLI,8,TITLD,0,DUMMY)
C
C     PLOT DATA
      CALL SETSMG(AMODE,30,2.00)
      CALL SETSMG(AMODE,45,1.25)
      DO 40 J=1,NPLOTS
      I = ND(J)
      IF(NPLOTS .EQ. 1) GO TO 41
      AI = DATA(NLFT,NI)
      AD = DATA(NLFT, I)
      CALL NUMERG(AMODE,AI,AD,1,J)
      CALL NUMERG(AMODE,AI,AD,1,J)
   41 NL = NLFT - 1
      NK = NP/NKP
      KK = 0
      DO 20 K=1,NP
      VI(K) = DATA(NL+K,NI)
      VD(K) = DATA(NL+K, I)
      IF(NPLOTS .EQ. 1) GO TO 20
      KK = KK + 1
      IF(KK .NE. NK) GO TO 20
      KK = 0
      CALL NUMERG(AMODE,VI(K),VD(K),1,J)
      CALL NUMERG(AMODE,VI(K),VD(K),1,J)
   20 CONTINUE
      CALL LINESG(AMODE,NP,VI,VD) -
      CALL LINESG(AMODE,NP,VI,VD)
      IF(NPLOTS .EQ. 1) GO TO 40
      AI = DATA(NRGT,NI)
      AD = DATA(NRGT, I)
      CALL NUMERG(AMODE,AI,AD,1,J)
      CALL NUMERG(AMODE,AI,AD,1,J)
   40 CONTINUE
C
C     ADVANCE FRAME
      CALL PAGEG(AMODE,0,1,1)
   45 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DYNS40
C                                                               DEBUG #  64
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 SAVEG, SAVEP, SAVED, SAVEA
      REAL*4 FMIN,FMAX,DBMIN,DBMAX,TITLE
      REAL*4 AMIN,AMAX
C
C     ---LATEST MOD ON 09/04/75
C
C  THIS OVERLAY PERFORMS THE LINEARIZED SYSTEM ANALYSES.
C
C
C
C         -----------DATA STREAM CONTROL-----------
C                    FOR THIS OVERLAY
C
C-----LNAM                                          FORMAT A4
C
C     IF (LNAM .EQ. 4H    ) RETURN
```

```
C       IF (LNAM .EQ. 4HTIME) GO TO 400
C
C-----CALL READIM (LRY,10,NCYC,10,KR)
C
C                     NOTE-----LRY(1,J) = ITYPE
C                              LRY(2,J) = ITFIN
C                              LRY(3,J) = JTFOUT
C                              LRY(4,J) = KPLOT
C                              LRY(5,J) = IAFLG
C                              LRY(6,J) = NO. B'S TO KEEP --ITYPE=7
C                              LRY(7,J) = LOCAL ID. NO. OF B'S TO RETAIN.
C                              LRY(8,J) = LOCAL ID. NO. OF B'S TO RETAIN.
C                              LRY(9,J) = LOCAL ID. NO. OF B'S TO RETAIN.
C
C
C
C-----CALL READIM (IRY,3,NCYC,3,KR)
C
C                     NOTE-----IRY(1,J) = ROOT TOLERANCE EXPONENT.
C                              IRY(2,J) = GAIN TOLERANCE EXPONENT
C                              IRY(3,J) = ROOT TOLERANCE EXPONENT
C                                          USED TO REMOVE SHIFT FREQ.
C                                          IN SUBROUTINE NUMS.
C
C       DO 500 ICYC = 1,NCYC
C
C-----TITLE                                        FORMAT (20A4)
C-----LPNAME,LPTAPE,LEIGV,LPOLY                    FORMAT (20A4)
C
C       IF(LEIGV.NE.4HEIGV) GO TO 23
C-----FQMIN,FQMAX                                  FORMAT(2F10.0)
C                IMAGINARY PART .GT. FQMIN AND .LT. FQMAX
C   23 CONTINUE
C
C       IF(LPOLY.NE.4HPOLY) GO TO 41
C          COMPUTE AND OUTPUT THE TRANSFER FUNCTION IN EXPANDED
C          POLYNOMIAL FORMAT
C   41 CONTINUE
C
C        IF(LPTAPE.EQ.4H7TRK) DUMP /PSTUFF/ ON 7-TRACK TAPE (UNIT 12)
C        CONTINUE
C
C       DO 500 ICP = 1,5
C
C
C       IF   (LPNAME(ICP) .EQ. 4H    ) GO TO 500
C       IF   (LPNAME(ICP) .EQ. 4HBODE
C     *.OR. LPNAME(ICP) .EQ. 4HNICH
C     *.OR. LPNAME(ICP) .EQ. 4HNYQU
C     *.OR. LPNAME(ICP) .EQ. 4HBLNN
C     *.OR. LPNAME(ICP) .EQ. 4HNINY) GO TO 200
C       IF   (LPNAME(ICP) .EQ. 4HROOT) GO TO 300
C
C       GO TO 9999
C
C  200 CONTINUE
C
C                -----BODE, NICHOLS, NYQUIST SECTION-----
C
C-----FMIN, FMAX, DBMIN, DBMAX, AMIN, AMAX          FORMAT (6F10.0)
```

```
C      GO TO 500                                                         0 6491
C                                                                        0 6492
C 300  CONTINUE                                                         0 6493
C              -----ROOT LOCUS SECTION-----                              0 6494
C                                                                        0 6495
C-----CALL [illegible] (IJM, 2, SRCC, 2, KR)                              0 6496
C-----CALL [illegible]  (RCC, 7, SRCC, KR, KR)                            0 6497
C                                                                        0 6498
C      DO 350 IHC=1,NREC                                                 0 6499
C                                                                        0 6500
C 350  CONTINUE                                                         0 6501
C      GO TO 500                                                         0 6502
C                                                                        0 6503
C 400  CONTINUE                                                         0 6504
C                                                                        0 6505
C 500  CONTINUE                                                         0 6506
C      RETURN                                                            0 6507
C                                                                        0 6508
C              -----DIMENSIONED WORK SPACE-----                          0 6509
C                                                                        0 6510
C                                                                        0 6511
       DIMENSION VS1(414),VS2(207)                                     412 6512
      DIMENSION EPNAME(3)                                                0 6513
      DIMENSION [illegible](2),GOOD(2)                                   0 6514
       DIMENSION RRN(100),RIN(100),RRD(100),RID(100),RZR(100),RZI(100) 414 6515
       DIMENSION IJM(2,100),ERY(5,100), IFY(3,100), KSKP(3)            418 6516
      DIMENSION ROOTRE(100),ROOTIM(100),A(100,4),IROJA(100,2)          415 6517
      DIMENSION AR(100),AI(100),VR(100),VI(100)                        416 6518
C                                                                        0 6519
C              -----COMMON BLOCKS-----                                   0 6520
C                                                                        0 6521
C                                                                        0 6522
       COMMON /KDSIZE/                                                   0 6523
     1                 KR, KRT, KRX, KV1, KV2, KVX                       0 6524
       COMMON /LDSIZE/                                                   0 6525
     2                 NX, NY, NDLTA, NKSS, N3, NJ2, NY2, ND2            0 6526
       COMMON /LCOUNT/                                                   0 6527
     3                 NNR, ICN, NNZ, NDR, ICD, NDZ                      0 6528
       COMMON /TAPENO/                                                   0 6529
     4                 NUT1, NUT2, NUT3                                  0 6530
       COMMON /MISCNO/                                                   0 6531
     5                 NOPRNT, NOPLOT                                    0 6532
       COMMON /LDDATA/                                                   0 6533
     6                 FMIN, FMAX, DBMIN, DBMAX                          0 6534
       COMMON /LTOL   /                                                  0 6535
     7                 TOLN,TOLD                                         0 6536
       COMMON /LTITLE/                                                   0 6537
     8                 TITLE(20)                                         0 6538
       COMMON /LIJV   /                                                  0 6539
     9                 IV(250)    , JV   (250)                         420 6540
       COMMON /LRARAY/                                                   0 6541
     A                 FBRN(100), FBNC(100), FBRD(100), FBDC(100)      422 6542
       COMMON /LROUT /                                                   0 6543
     B                 R(207)     , RX   (414)                         424 6544
       COMMON /  LV1 /                                                   0 6545
     C                 V1   (100), V2   (100), V3   (100)              426 6546
       COMMON /  LV2 /                                                   0 6547
     D                 XV1 (100), XV2 (100), XV3 (100), XV4 (100)      428 6548
       COMMON /VECTOR/                                                   0 6549
     E                 Y    (250), YD   (250)                          430 6550
```

```
       COMMON /LWORK1/                                                      0 6551
      F               W1(100,100), W2(100,100)                            432 6552
       COMMON /TIMESS/                                                      0 6553
      C               ST, DT, T, ET, TMST                                   0 6554
       COMMON /PLTDTA/                                                      0 6555
      1                NRPLOT,NCPLOT                                        0 6556
       COMMON /FSTUFF/                                                      0 6557
      *             SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE 447 6558
C                                                                           0 6559
C                                                                           C 6560
C         -----EQUIVALENCE MAP-----                                         0 6561
C                                                                           0 6562
C                                                                           C 6563
C                                                                           0 6564
       EQUIVALENCE (VS1(208),VS2(1))                                      434 6565
C                                                                           0 6566
C                                                                           0 6567
C         -----DATA STATEMENTS-----                                         0 6568
C                                                                           C 6569
C                                                                           0 6570
       DATA  LBODE, LROOT, LNYQU, LNICH, LNINY, LTIME, LBLNK, LBONN/        0 6571
      *      4HBODE,4HROOT,4HNYQU,4HNICH,4HNINY,4HTIME,4H    ,4HBONN/       0 6572
C                                                                           0 6573
       DATA NIT/ 5/                                                         0 6574
       DATA NOT/ 6 /                                                        0 6575
       DATA IFST,NOT12,L7IRK/0,12,4H7IRK/                                   0 6576
       DATA LEGVAL/4HEIGV/                                                  0 6577
       DATA LPOLYN/4HPOLY/                                                  0 6578
C                                                                           0 6579
       LOGICAL DEBUG,LEGO                                                   C 6580
       COMMON /LDEBUG/ DEBUG(120)                                           0 6581
       EQUIVALENCE (LEGO,DEBUG(64))                                         0 6582
C                                                                           0 6583
 1003 FORMAT (20A4)                                                         0 6584
 1004 FORMAT (8F10.0)                                                       0 6585
 2001 FORMAT (//,5X,10HON ICYC = ,I2,26H,NUMERATOR GAIN LESS THAN ,         C 6586
      *           D10.4,16HWAS ENCOUNTERED.,//,5X,                          0 6587
      *                 23HTHE BODE GAIN VALUE IS D10.4,//,5X,              0 6588
      *           47HPROGRAM CONTINUING WITH NEXT TRANSFER FUNCTION.)       0 6589
C                                                                           0 6590
       IF(LEGO) WRITE(NOT,1000)                                             C 6591
 1000 FORMAT ('  SUBROUTINE DYNS40 ')                                       0 6592
C                                                                           0 6593
C         -----SET UP FIXED INTEGER DATA-----                               0 6594
C                                                                           0 6595
      KR  = 100                                                           436 6596
      KRT = 207                                                           438 6597
      KRX = 414                                                           440 6598
      KV1 = 414                                                           442 6599
      KV2 = 207                                                           444 6600
      KVX = 100                                                           446 6601
      K3 = 3                                                                0 6602
      K9 = 9                                                                0 6603
C                                                                           0 6604
C         -----SET UP VARIABEL INTEGER DATA-----                            0 6605
C                                                                           0 6606
      KY2 = NX - NDLTA - NXSS                                               0 6607
      ND2 = NDLTA - NB                                                      C 6608
C                                                                           0 6609
      REWIND NOT2                                                           0 6610
      REWIND NUT3
C
C          READ IN LINEARIZED PARTIAL DERIVATIVE MATRIX FROM UNIT NUT2
C          PERFORM SIMILARITY TRANSFORMATION, AND PUT A* BACK ON NUT2.
C
C        FROM SUBROUTINE CONTRL:
C             NXSS = NUMBER OF SENSOR SIGNALS
C             NB = NUMBER OF TORQUE SIGNALS
C             NX, NJG COMPUTED IN LINEAR
C             R = ZNEW VECTORS COMPUTED IN LINEAR
C             W1 = THE H MATRIX OF EQUATION III-21 VOL 1
C             W2 = THE C MATRIX OF EQUATION III-20 VOL 1
C
C
      NAUX = NXSS + NB
      DO 50 L=1,NX
      READ (NUT2) (R(I),I=1,NJG)
      DO 45 I=1,NX
   45 W1(I,L) = R(I)
C
      DO 46 I=1,NAUX
      J = NX+I
   46 W2(I,L) = R(J)
   50 CONTINUE
      IF(LEQU) WRITE (NOT,1001)
 1001 FORMAT (' -A- IS THE COEFFICIENT MATRIX FOR THE LINEARIZED PERTURB
     *ATION EQUATIONS ')
      CALL WRITE (W1,NX,NX,3H-A-,KR)
      IF(LEQU) WRITE (NOT,1005)
 1005 FORMAT (' -A-AUX IS THE COEFFICIENT MATRIX USED TO DEFINE SENSOR A
     *ND TORQUE SIGNALS AS LINEAR FUNCTIONS OF STATE VARIABLES ')
      CALL WRITE (W2,NAUX,NX,6H-A-AUX,KR)
C
      REWIND NUT2
      WRITE (NUT2) ((W1(I,J),I=1,KR),J=1,KR)
      WRITE (NUT2) ((W2(I,J),I=1,KR),J=1,KR)
C
      REWIND NUT2
C        GET COMPLEX ROOTS OF THE LINEARIZED COUPLED SYSTEM EQUATIONS
C        PLANT PLUS CONTROL SYSTEM EQUATIONS
      CALL SRGRVR (W1,NX,RRD,RID,KR)
      READ (NUT2)((W1(I,J),I=1,KR),J=1,KR)
      READ (NUT2) ((W1(I,J),I=1,KR),J=1,KR)
      REWIND NUT2
C
C        W1 CONTAINS THE COEFFICIENT MATRIX FOR THE SENSOR AND TORQUE
C        SIGNALS
      CALL ASIMLR (W1,W2,IV,KR)
C        SAVE W2 = R**(-1)*H*R OF EQUATION III-2, VOL 1 REORDED FORM
      WRITE (NUT2) ((W2(I,J),I=1,KR),J=1,KR)
      REWIND NUT2
      IF(LEQU) WRITE(NOT,1002)
 1002 FORMAT (' -A*- IS THE COEFFICIENT MATRIX FOR THE LINEARIZED EQUATI
     *ONS OF MOTION WITH SENSOR AND TORQUE SIGNALS ')
      CALL WRITE (W2,NX,NX,4H-A*-,KR)
C        GET COMPLEX ROOTS OF TRANSFORMED EQUATIONS
      CALL SRGRVR (W2,NX,RRN,RIN,KR)
C  REMOVE NUMBERS SMALLER THAN 1.D-5 FROM ROOT ARRAYS.
C
      CALL SIFT (RID,NX,1.D-5)
      CALL SIFT (RID,NX,1.D-5)
      CALL SIFT (RRN,NX,1.D-5)
      CALL SIFT (RIN,NX,1.D-5)
C
C          PRINT BOTH SETS OF COMPLEX ROOTS. THEY SHOULD BE EQUAL
C          THEIR CLOSENESS IS A QUALITITY MEASURE FOR THE
C          SIMILARITY TRANSFORMATION
      CALL RWRITE (2,RRD,RID,RRN,RIN,NX,NX,4HRT A,4HRTA*)
      READ (NCT2)((W2(I,J),I=1,KR),J=1,KR)
      REWIND NCT2
C
C  READ IN CONTROL VARIABLE. LNAM AND BRANCH TO APPROIATE SECTION.
C******************* READ IN TYPE OF  STABILITY ANALYSIS
      READ (NIT,1003) LNAM
C          LNAM = 4HTIME LINEARIZED TIME RESPONSE
C               = 4HFREQ FREQUENCY RESPONSE ANALYSIS
C               = BLANK RETURN
C
      IF (LNAM .EQ. LBLNK) RETURN
      IF (LNAM .EQ. LTIME) GO TO 400
C
C               -----FREQUENCY DOMAIN ANALYSIS SECTION-----
C
C
C  READ IN FREQUENCY ANALYSIS CONTROL VARIABLES
C******************* READ SPECIFICATIONS TO DEFINE TRANSFER FUNCTIONS
C          LRY  = ARRAY DEFINES TRANSFER FUNCTION SPECIFICATION DATA
C          NCYC = NUMBER OF SEPERATE TRANSFER FUNCTIONS TO CONSIDER
      CALL READIM (LRY,N9,NCYC,K9,KR)
                                                           NERROR = 0
C
C******************* READ EXPONENT TOLERANCE DATA
      CALL READIM (IRY,N3,NC2,K3,KR)
C
      IF (NC2 .NE. NCYC) GO TO 9999
C
      DO 500 ICYC = 1,NCYC
C
      ITYPE = LRY(1,ICYC)
      ITFIN = LRY(2,ICYC)
      JTFOUT = LRY(3,ICYC)
      KPLOT = LRY(4,ICYC)
      IAPLC = LRY(5,ICYC)
C          ITYPE = TRANSFER FUNCTION TYPE
C          ITFIN = INTEGER TO DEFINE INPUT SIGNAL FOR TRANSFER FUNCTION
C          JTFOUT = INTEGER TO DEFINE OUTPUT SIGNAL
C          KPLOT = INTEGER TO DEFINE IF PLOTS REQUIRED
C          IAPLC = SHOULD CHARATERISTIC EQUATION OR ITS TRANSPOSE BE USED
C                  TO GET TRANSFER FUNCTION
                                                           NERROR = 1
      IF (ITYPE .EQ. 3) GO TO 300
C
      IF (IABS(ITYPE) .LE. 6) GO TO 55
      NBKP = LRY(6,ICYC)
      DO 54 I=1,NBKP
   54 KBKP(I) = LRY(I+6,ICYC)
   55 CONTINUE
C
C
C  SET UP DEFAULT VALUES IN ARRAY IRY.
```

```
C
      IF (IRY(1,ICYC) .EQ. 0) IRY(1,ICYC) = -7
      IF (IRY(2,ICYC) .EQ. 0) IRY(2,ICYC) = -7
      IF (IRY(3,ICYC) .EQ. 0) IRY(3,ICYC) = -7
C
      TOLN = 10.D0 ** IRY(1,ICYC)
      CTOL = 10.D0 ** IRY(2,ICYC)
      FTOL = 10.D0 ** IRY(3,ICYC)
C
      TOLD = TOLN
C
C
      IF (IABS(ITYPE) .LT. 1  .OR.  ITYPE .GT. 5) GO TO 9999
C
C******************** READ TITLE FOR TRANSFER FUNCTION ICYC
      READ (NIT,1003) TITLE
C
      READ (NUT2) ((W2(I,J),I=1,KR),J=1,KR)
C        W2 = R**(-1)*H*R IN EQUATION III-24 VOL 1
      REWIND NUT2
C
      CALL TFTYPE (W2,W1,V1,NX,NA,ITYPE,ITFIN,NISP,CJSP,KR,KR)
C        EXIT TFTYPE WITH
C              W1 = CHARACTERISTIC MATRIX
C              V1 = CORRESPONDING INPUT COEFFICIENTS
C        ASSOCIATED WITH PARTICULAR TYPE TRANSFER FUNCTION
C        PUT CHARACTERISTIC MATRIX ON NUT3
      CALL WRITE (W1,NA,NA,4H-AR-,KR)
      CALL WRITE (V1,1,NA,4HBCOL,1)
      WRITE (NUT3) ((W1(I,J),I=1,KR),J=1,KR)
      REWIND NUT3
C
C  NOW HAVE REDUCED TF A* ON NUT3.
C
C  OBTAIN ROOTS FOR DENOMINATOR.
C
C
      CALL QRQRVR (W1,NA,RRD,RID,KR)
      CALL SIFT (RRD,NA,TOLD)
      CALL SIFT (RID,NA,TOLD)
C
      READ (NUT3) ((W2(J,I),I=1,KR),J=1,KR)
      REWIND NUT3
C
C        GET ROOTS OF TRANSPOSE OF CHARACTERISTIC EQUATION
      CALL QRQRVR (W2,NA,V2,V3,KR)
      CALL SIFT (V2,NA,TOLD)
      CALL SIFT (V3,NA,TOLD)
      CALL RWRITE (2,RRD,RID,V2,V3,NA,NA,4HR AR,4HMAXI)
C
      IF (IAFLG .EQ. 0) GO TO 59
      DO 58 I=1,NA
      RRS(I) = V2(I)
   58 RIS(I) = V3(I)
   59 CONTINUE
C
C  OBTAIN ROOTS OF NUMERATOR.
C
      READ (NUT3) ((W1(I,J),I=1,KR),J=1,KR)
      REWIND NUT3
```

```
C         LOCATE OUTPUT SIGNAL FOR COMPUTATION OF PARTICULAR TYPE
C         TRANSFER FUNCTION CALLED FOR
C   OBTAIN ABSOLUTE JTF LOCATION BASED UPON LOCAL JTFOUT.
      JTF = NY2 + JTFOUT
      IF (IABS(ITYPE) .EQ. 2) JTF = ND2 + JTFOUT
      IF (IABS(ITYPE) .EQ. 5) JTF = JTF + NXSS + ND2
      IF (IABS(ITYPE) .EQ. 7) JTF = JTF + NXSS + ND2
      IF (IABS(ITYPE) .EQ. 8) JTF = JTFOUT
      CALL NUMS (W1,W2,V1,RRN,RIN,R2R,R2I,PTOL,
     *            GAIN,IFLG,NNUM,NZRO,JTF,NA,KR)
C
      NN = NZRO
      ND = NA
C
      IF (IFLG .NE. 0) GO TO 65
   64 IFLG = 0
      CALL PAGEHD
      WRITE (NOT,2001) ICYC,GTOL,GB
      GO TO 75
   65 CONTINUE
C
      CALL SIFT (RRN,NN,IGEN)
      CALL SIFT (RIN,NN,IGEN)
C         CALL OCQRRT TO SEPERATE NUMERATOR QR ROOTS INTO REAL AND ZERO
C         ROOTS (RETURNED IN V1) AND COMPLEX PAIRS (RETURNED IN V2)
      CALL OCQRRT (RRN,RIN,NN,NNR,ICN,NNZ,V1,V2)
      CALL DFCRMB (NNR,ICN,V1,V2,FBRN,FBNC,V3,1.00,GAIN,GNB)
C
   70 CONTINUE
C
      CALL RWRITE (2,RRN,RIN,RRD,RID,NN,ND,4H NUM,4H DEN)
C
C         SEPERATED QR ROOTS OF DENOMINATOR
      CALL OCQRRT (RRD,RID,ND,NDR,ICD,NDZ,V1,V2)
      CALL DFCRMB (NDR,ICD,V1,V2,FBRD,FBDC,V3,1.00,1.00,GDB)
      GB = GNB(2)/GDB(2)
C         GB = BODE GAIN SEE EQUATION III-48 AND III-70, VOL 1
C         ARRAYS FBRN, FBNC, FBRD, AND FBDC CONTAIN SYSTEM TIME CONSTANTS
C         DAMPING AND FREQUENCIES REQUIRED IN EQUATION III-70, VOL 1
      IF (DABS(GB) .LT. GTOL) GO TO 64
C
C-----ESTABLISH WORKING ROOT COUNTS.
C
      KNR = NNR
      KDR = NDR
      KNZ = NNZ
      KDZ = NDZ
      KCN = ICN
      KCD = ICD
C
      CALL ZERO (XV1,1,KVX,1)
      CALL ZERO (XV2,1,KVX,1)
      CALL ZERO (XV3,1,KVX,1)
      CALL ZERO (XV4,1,KVX,1)
C
      IF (KNR .EQ. 0) GO TO 2202
      DO 201 I=1,KNR
  201 XV1(I) = FBRN(I)
 2202 CONTINUE
      IF (KDR .EQ. 0) GO TO 2203
      DO 202 I=1,KDR
  202 XV3(I) = FBRD(I)
 2203 CONTINUE
      IF (KCN .EQ. 0) GO TO 2204
      K=2*KCN
      DO 203 I=1,K
  203 XV2(I) = FBNC(I)
 2204 CONTINUE
      IF (KCD. EQ. 0) GO TO 205
      K=2*KCD
      DO 204 I=1,K
  204 XV4(I) = FBDC(I)
  205 CONTINUE
C
C-----REMOVE REAL ZEROS PRIOR TO CALL TO ITFF
      IF (KNR .NE. 0) CALL RMVZRO (XV1,KNR)
      IF (KDR .NE. 0) CALL RMVZRO (XV3,KDR)
      CALL ZERO (R,1,KRT,1)
C
C          CALL ITFF TO SET UP THE ROOTS ARRAY R
      CALL ITFF (KNR,KCN,KNZ,KDR,KCD,KDZ,
     *           GB,XV1,XV2,XV3,XV4,R,KRT)
C
C          CALL CANCOR TO CANCEL OUT ROOTS COMMON TO NUM. AND DEN.
      CALL CANCOR (R)
      IF(.NOT.LEQU) GO TO 10
      WRITE(NOT,1006)
 1006 FORMAT (' ROOT ARRAY OUT OF CANCOR IS ')
      CALL WRITES(R,1,KRT,1)
   10 CONTINUE
C
C
C******************** READ IN DISPLAY CONTROL VARIABLES
C
   75 READ(NIT,1003) LPNAME,LPTAPE,LEIGV,LPOLY
C
C          CHECK TO SEE IF EIGENVECTORS ARE TO BE COMPUTED
C              READ FREQUENCY SPREAD FOR ROOTS OF CHARACTERISTIC EQUATION
      IF(LEIGV.NE.LEGVAL) GO TO 25
C
C********************READ IN RANGE FOR EIGENVECTOR COMPUTATION
C
      READ(NIT,1004) FGMIN,FGMAX
      IF(IFLG.EQ.0) GO TO 23
      IC = 0      \
      DO 24 I=1,ICD
      IF(V2(2*I).LT.FGMIN.OR.V2(2*I).GT.FGMAX) GO TO 24
      IC = IC+1
      ROOTRE(IC) = V2(2*I-1)
      ROOTIE(IC) = V2(2*I)
   24 CONTINUE
C
C          PUT ROOT IN ORDER
      DO 26 M=1,IC
      VSV = ROOTIE(M)
      ISV = M
      M1 = M+1
      IF(M1.GT.IC) GO TO 28
      DO 27 N=M1,IC
      IF(VSV.GE.ROOTIE(N)) GO TO 27
      VSV = ROOTIE(N)
      ISV = N
   27 CONTINUE
   28 CONTINUE
      WSV = ROOTRE(ISV)
      ROOTIE(ISV) = ROOTIE(M)
      ROOTRE(ISV) = ROOTRE(M)
      ROOTIE(M) = VSV
      ROOTRE(M) = WSV
   26 CONTINUE
C
      CALL WRITE(ROOTRE,1,IC,6HROOTRE,1)
      CALL WRITE(ROOTIE,1,IC,6HROOTIE,1)
      CALL PAGEND
      WRITE(NOT,1010)
 1010 FORMAT (///,10X,'REFER TO THE TRANSFORMED STATE VECTOR CORRELATION
     * ARRAY AND CODE ITYPE NUMBERING LOGIC IN SUBROUTINE IFTYPE',/,
     * 10X,'TO ASSOCIATE EACH EIGENVECTOR ELEMENT WITH A PARTICULAR DEGR
     *EE OF FREEDOM')
C
      DO 25 M=1,IC
      I = IC+1-M
      READ (NOT3) ((W1(II,J),II=1,KR),J=1,KR)
      REWIND NOT3
      CALL ASGR(0,W1,W2,NA,KR)
      CALL EIGVEC(3,W1,W2,W,IROW,XR,XI,VR,VI,ROOTRE(I),ROOTIE(I),NA,KR,0
     *,SW1,COUNTE,ERR)
      WRITE (NOT,1007) ROOTRE(I),ROOTIE(I)
 1007 FORMAT (///,10X,'EIGENVALUE = ',D15.8,' + I*',D15.8,//)
      WRITE (NOT,1008)
 1008 FORMAT (10X,'EIGENVECTOR OF CHARACTERISTIC MATRIX -AR-',16X,
     *  'EIGENVECTOR OF TRANSPOSE OF CHARACTERISTIC MATRIX -AR**T-',//)
      WRITE(NOT,1009) (J,XR(J),XI(J),J,VR(J),VI(J),J=1,NA)
 1009 FORMAT (5X,I5,10X,D15.8,' + I*(',D15.8,')',
     *           5X,I5,10X,D15.8,' + I*(',D15.8,')')
      WRITE(NOT,1011)
 1011 FORMAT (/,10X,'SAME EIGENVECTOR EXPRESSED IN POLAR COORDINATES WIT
     *H PHASE GIVEN IN DEGREES (MAG,PHASE)',/)
      DO 11 J=1,NA
      XMG = DSQRT(XR(J)**2 + XI(J)**2)
      VMG = DSQRT(VR(J)**2 + VI(J)**2)
      IF(XMG.EQ.0.0D0) GO TO 12
      XAG = DATAN2(XI(J),XR(J))*57.2957795D+00
      XR(J) = XMG
      XI(J) = XAG
   12 IF(VMG.EQ.0.0D0) GO TO 11
      VAG = DATAN2(VI(J),VR(J))*57.2957795D+00
      VR(J) = VMG
      VI(J) = VAG
   11 CONTINUE
      WRITE(NOT,1012) (J,XR(J),XI(J),J,VR(J),VI(J),J=1,NA)
 1012 FORMAT (5X,I5,10X,'(',D15.8,' , ',D15.8,')',
     *           7X,I5,10X,'(',D15.8,' , ',D15.8,')')
   25 CONTINUE
C
   23 CONTINUE
C          CHECK TO SEE IF 7 TRACK TAPE OF /PSTUFF/ IS TO BE WRITTEN
      IF(LPTAPE.NE.L7TRK) GO TO 20
      IF(IFST.NE.0) GO TO 20
      IFST = 1
      REWIND NUT12                                                      C 6971
  [illegible] CONTINUE                                                  0 6972
C                                                                       0 6973
      DO 500 ICP = 1,5                                                  C 6974
C                                                                       C 6975
      IF   (LPNAME(ICP) .EQ. LBLNK ) GO TO 500                          0 6976
      IF   (LPNAME(ICP) .EQ. LBODE                                      0 6977
     *.OR. LPNAME(ICP) .EQ. LNICH                                       C 6978
     *.OR. LPNAME(ICP) .EQ. LNYQU                                       0 6979
     *.OR. LPNAME(ICP) .EQ. LUCNN                                       0 6980
     *.OR. LPNAME(ICP) .EQ. LNINY ) GO TO 200                           0 6981
      IF   (LPNAME(ICP) .EQ. LROOT ) GO TO 300                          0 6982
                                                          NERROR = 2    C 6983
      GO TO 9999                                                        0 6984
C                                                                       0 6985
  200 CONTINUE                                                          C 6986
C                                                                       0 6987
C                -----FREQUENCY RESPONSE PROCESSING-----                C 6988
C                                                                       0 6989
C***************** READ IN FREQUENCY RESPONSE BANDS FOR COMPUTATION     0 6990
      READ (NIT,1004) FMIN, FMAX, DBMIN, DBMAX, AMIN, AMAX              0 6991
      IF (IFLG .EQ. 0) GO TO 500                                        C 6992
C                                                                       0 6993
      KNR = R(1) + 0.100                                                0 6994
      KCN = R(2) + 0.100                                                0 6995
      KNZ = R(3) + 0.100                                                0 6996
      KDR = R(4) + 0.100                                                0 6997
      KCD = R(5) + 0.100                                                0 6998
      KDZ = R(6) + 0.100                                                0 6999
      CALL WRITE (R,1,KRT,NHRREC,1)                                     C 7000
C                                                                       0 7001
      CALL ZERO (XV1,1,KVX,1)                                           0 7002
      CALL ZERO (XV2,1,KVX,1)                                           C 7003
      CALL ZERO (XV3,1,KVX,1)                                           C 7004
      CALL ZERO (XV4,1,KVX,1)                                           C 7005
C                                                                       C 7006
C         READ ROOT DATA INTO                                           0 7007
C               XV1 = NUMERATOR REAL                                    0 7008
C               XV2 = NUMERATOR COMPLEX                                 C 7009
C               XV3 = DENOMINATOR REAL                                  0 7010
C               XV4 = DENOMINATOR COMPLEX                               0 7011
      IF (KNR .EQ. 0) GO TO 2207                                        0 7012
      DO 206 I=1,KNR                                                    0 7013
      L=7+I                                                             0 7014
  206 XV1(I) = R(L)                                                     0 7015
 2207 CONTINUE                                                          0 7016
      IF (KCN .EQ. 0) GO TO 2208                                        0 7017
      K=2*KCN                                                           0 7018
      DO 207 I=1,K                                                      0 7019
      L=7+KNR+I                                                         C 7020
  207 XV2(I) = R(L)                                                     0 7021
 2208 CONTINUE                                                          C 7022
      IF (KDR .EQ. 0) GO TO 2209                                        C 7023
      DO 208 I=1,KDR                                                    C 7024
      L = 7+KNR+2*KCN+I                                                 0 7025
  208 XV3(I) = R(L)                                                     C 7026
 2209 CONTINUE                                                          0 7027
      IF (KCD .EQ. 0) GO TO 2210                                        0 7028
      K=2*KCD                                                           C 7029
      DO 209 I=1,K                                                      C 7030
      L=7+KNR+2*KCN+KDR+I
  205 XV4(I) = R(L)
 2210 CONTINUE
C
C-----EXTEND REAL ARRAY COUNTS TO INCLUDE REAL ZEROS.
      KNR = KNR + KNZ
      KDR = KDR + KDZ
C
C-----PERFORM THE FREQUENCY RESPONSE.
C
      CALL SFREQ2 (KNR,KCN,KDR,KCD,GB,
     1                  XV1,XV2,XV3,XV4,FMIN,FMAX,TITLE)
C
C         CHECK TO SEE IF REQUEST MADE TO COMPUTE POLYNOMIAL FORM
C         OF TRANSFER FUNCTION FOR OUTPUT
      IF(LPOLY.NE.LPOLYN) GO TO 41
      CALL PAGEHD
      CALL RICF(R,VS1,VS2,KRT)
      WRITE(NOT,38)
   38 FORMAT('1P(S) =')
      NP1 = VS1(1)
      WRITE(NOT,40) VS1(3),(VS1(I+3),I,I=1,NP1)
      WRITE(NOT,39)
   39 FORMAT('0Q(S) =')
      NQ1 = VS1(2)
      WRITE(NOT,40) VS1(NP1+4),(VS1(I+NP1+4),I,I=1,NQ1)
   40 FORMAT (3X,D22.15,7X,3(1X,'+(',D22.15,'*S**',I2,')')/,
     *                    4(1X,'+(',D22.15,'*S**',I2,')'))
   41 CONTINUE
C
C
      IF(LPTAPE.NE.L7TRK) GO TO 21
      WRITE(NOT12,100) IFST,KSAVE
      WRITE (NOT12,101) (SAVEG(I),SAVEP(I),SAVED(I),SAVEA(I),I=1,KSAVE)
      WRITE(NOT,103)
      WRITE(NOT,102) TITLE,LPNAME(IOP),IFST
      IFST = IFST+1
  100 FORMAT (2I5)
  101 FORMAT (4E18.8)
  102 FORMAT (20A4,/,3X,A4,' IN FILE',I4,' OF DATA SET 12')
  103 FORMAT (//,'  FREQUENCY RESPONSE DATA FOR')
  104 FORMAT (//,' ROOT LOCUS DATA FOR ')
   21 CONTINUE
C
      IF (KPLOT .EQ. 0) GO TO 220
      IF (LPNAME(IOP) .EQ. LBODE
     * .OR.LPNAME(IOP) .EQ. LBONN)
     *                    CALL SPLOT (TITLE,FMAX,FMIN,DBMIN,DBMAX)
C
      IF  (LPNAME(IOP) .EQ. LNICH
     * .OR.LPNAME(IOP) .EQ. LNINY
     *.OR.LPNAME(IOP) .EQ. LBONN) CALL NIPLOT (TITLE,DBMIN,DBMAX)
C
      IF  (LPNAME(IOP) .EQ. LNYQU
     * .OR.LPNAME(IOP) .EQ. LBONN
     * .OR.LPNAME(IOP) .EQ. LNINY) CALL NYPLOT (TITLE,AMIN,AMAX)
C
      GO TO 300
  220 CALL PAGEHD
      WRITE (NOT,221) ICYC, LPNAME(IOP)
  221 FORMAT (//,10X,
     *  48HNO FREQUENCY RESPONSE PLOTS GENERATED ON ICYC = I3,
     *  //,10X,9HLPNAME = A4)
      GO TO 500
  300 CONTINUE
C
C               -----ROOT LOCUS SECTION-----
C
C
      CALL RTCP (R,RX,V51,KRX)
      NP = RX(1) + 1.D0
      NQ = RX(2) + 1.D0
C
      DO 320 I=1,NP
      J=I+2
  320 XV1(I) = RX(J)
      DO 325 I=1,NQ
      J = I+2+NP
  325 XV2(I) = RX(J)
      CALL WRITE (XV2,1,NQ,4HPDEN,1)
      CALL WRITE (XV1,1,NP,4HPNUM,1)
C
C
C     READ IN ROOT LOCUS CONTROL VARIABLES.
C
C******************** READ IN ROOT LOCUS CONTROL DATA
  330 CALL READIM (IJM,NR2,NRLC,2,KR)
      IF (IFLG .EQ. 0) GO TO 340
                                                                  NERROR = 3
      IF (NR2 .NE. 2  .OR.  NRLC .GT. KR) GO TO 9999
C
C--------NOTE..... IJM(1,J) = ISKIM(J)
C                   IJM(2,J) = ELE. LOCATION IN ROOT ARRAY
C                              FOR STARTING ROOT LOCUS.
C
C******************** READ IN ADDITIONAL ROOT LOCUS DATA
  340 CALL READ (W1,NR2,NC2,KR,KR)
      IF (IFLG .EQ. 0) GO TO 500
                                                                  NERROR = 4
      IF (NR2 .NE. 6  .OR.  NC2 .NE. NRLC) GO TO 9999
C
C-----NOTE.....W1(1,J) = THETAC(J)
C              W1(2,J) = SCL
C              W1(3,J) = ALOC
C              W1(4,J) = XMIN
C              W1(5,J) = XMAX
C              W1(6,J) = YMAX
C
      DO 350 IRC = 1,NRLC
C
C
      ISKIM  = IJM(1,IRC)
      JJ     = IJM(2,IRC)
C
      THETAC = W1(1,IRC)
      SCL    = W1(2,IRC)
      ALOC   = W1(3,IRC)
      XMIN   = W1(4,IRC)
      XMAX   = W1(5,IRC)
      YMAX   = W1(6,IRC)
```

```
C
C     LOCATE PROPER STARTING ROOT.
C
      IF (ISNIM .NE. 1) GO TO 341
C
C     ROOT IS AN OPEN LOOP ZERO.
C
      SR = RRN(JJ)
      SI = RIN(JJ)
      GO TO 342
  341 CONTINUE
C
C     ROOT IS A POLE.
C
      SR = RRD(JJ)
      SI = RID(JJ)
  342 CONTINUE
C
      CALL RLOCUS (XV1,XV2,SCL,SR,SI,NP,NQ,THETAJ,
     1                 XMIN,XMAX,YMAX,ALOC)
C
      IF(LFTAPE.NE.L7TRK) GO TO 22
      WRITE(NOT12,100) IFST,KSAVE
      WRITE(NOT12,101) (SAVEL(I),SAVEP(I),SAVED(I),SAVEA(I),I=1,KSAVE)
      WRITE(NOT,104)
      WRITE(NOT,102) TITLE,LPNAME(IOP),IFST
      IFST = IFST+1
   22 CONTINUE
C
      IF (KPLOT .EQ. 1) CALL REPLOT (TITLE,ISNIM,ICYC,IRC)
C
  350 CONTINUE
C
C
      GO TO 500
  400 CONTINUE
C
C              -----LINEARIZED TIME RESPONSE SECTION-----
C
      READ (NOT2) ((W1(I,J),I=1,KR),J=1,KR)
      REWIND NOT2
      CALL LTRESP
      RETURN
C
  500 CONTINUE
C
C
      RETURN
C
 9999 WRITE (NOT,1999) NERROR
 1999 FORMAT (1H1,//10X,44HERROR ENCOUNTERED IN OVERLAY(4,0). NERROR = ,
     *        I3,/10X,16HPROGRAM STOPPED.)
      STOP
C
      END
      SUBROUTINE ASIMLR (A,B,IV,KR)
C                                                              DEBUG #  65
      IMPLICIT REAL*8(A-H,O-Z)
C
C     ---LATEST MOD ON 05/04/75
C
C   SUBROUTINE ESTABLISHES TRANSFORMED PARTIAL DERIVATIVE MATRIX
C              BY PERFORMING A SIMILARITY TRANSFORMATION TO
C              EXCHANGE PLANT STATE VARIABLES ,Y, FOR SENSOR SIGNALS,XSS
C              AND CONTROL SYSTEM VARIABLES ,DELTA, FOR TORQUE
C              VARIABLES ,B.
C
C   THE VARIABLE SEQUENCE IS REORDERED FROM Y,DELTA,XSS,B
C                                         TO Y,XSS,DELTA,B
C
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C   A       = INPUT  MATRIX OF PARTIAL DERIVATIVES.  COORDINATE ORDER
C                    IS XSS, B. SIZE IS (NJQ-NX),NX.
C   B       = OUTPUT  TRANSFORMED AND REORDERED PARTIAL DERIVATIVE
C                    MATRIX.  ORDER IS Y,XSS,DELTA,B.  SIZE IS NX,NX.
C   IV      = INPUT  INTEGER WORK VECTOR.  SIZE MUST BE AT LEAST NX.
C   KR      = INPUT  ROW DIMENSION SIZE OF A AND B IN CALLING PROGRAM.
C
      DIMENSION A(KR,1), B(KR,1), IV(1)
C
      COMMON /LDSIZE/
     2              NX, NY, NDELTA, NXSS, NIT3, NJQ, NY2, ND2
          COMMON /SPECIF/
     *      BETAH(6, 5),BETAHD(6, 6),AMO(2, 5),RH(3,3,30),RS(3,3,30),
     *      BH(3,30),BS(3,30),IMO(3, 5),NMOM(6, 5),IFFSMA(15),
     *      NS,NH,NSWT,NORMC,NDELTA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6),
     *      LOCL(14),LENOL(14),NG,NBETA,NLAM,NEQ
      COMMON /TAPENO/
     4              NOT1, NOT2, NOT3
      COMMON /VECTOR/
     5              Y   (250), YD  (250)
          COMMON /VINDEP/
     *      INDEP(250)
      DATA NOT/6/
      LOGICAL DEBUG,LEQU
      COMMON /LDEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(65))
C
C   SET UP C.  LOWER PART OF -A- REQUIRED TO OBTAIN -T-.
C
      NR = NJQ - NX
      NC = NJQ
      KC = (3*KR)/5
      IF(LEQU) WRITE(NOT,1000)
 1000 FORMAT (' SUBROUTINE ASIMLR ')
      IF(.NOT.LEQU) GO TO 50
      WRITE (NOT,1001)
 1001 FORMAT (' INPUT MATRIX COEFFICIENTS FOR LINEARIZED SIGNAL EQUATION
     *S ')
      CALL WRITES(A,NR,NX,KR)
   50 CONTINUE
C     THE USE OF KC PERMITS THE B SPACE TO BE BETTER UTILIZED
C     VIA EXTENDING THE PERMISSABLE NO. OF AUXILLARY EQUATIONS.
C
```

```
      CALL ZERO (B,KR,KR,KR)
C     SET UP B SPACE FOR KC ROW DIMENSION WITH AUGMENTED A.
C
      DO 10 I=1,NR
      L = I+NX
      DO 10 J=1,NX
C
      IJ2 = I+KC*(J-1)
      J2  = (IJ2-1)/KR+1
      I2  = IJ2-KR*(J2-1)
C
      B(I2,J2) = A(I,J)
C
      IF (I .NE. J) GO TO 10
C
      IJL2 = I+KC*(L-1)
      JL2  = (IJL2-1)/KR+1
      IL2  = IJL2-KR*(JL2-1)
C
      B(IL2,JL2) = -1.D0
C
   10 CONTINUE
C
C          THE ARRAY B NOW CONTAINS EQUATION III-27 VOL 1
C   ESTABLISH SEARCH LIMIT FOR SUBROUTINE FINDT
C
      NS = NX
      CALL FINDT (B,NR,NC,NS,A,NRET,KC,KR)
C          EXIT FINDT WITH THE SIMILARITY TRANSFORMATION MATRIX R
C          OF EQUATION III-23 VOL 1, STORED IN A ARRAY
      CALL WRITE (A,NRET,NRET,4H-T- ,KR)
C
C   A = -T-
C
C   FORM -A*- = T(INV) A T
C
C          READ PARTIAL DERIVATIVE MATRIX H
      READ (NUT2)  ((B(I,J),I=1,KR),J=1,KR)
C          PUT SIMILARITY TRANSFORMATION MATRIX R ON NUT2 AFTER H
      WRITE (NUT2)  ((A(I,J), I=1,KR),J=1,KR)
      REWIND NUT2
C   INVERT -T- USING GAUSSI (THE SIMILARITY TRANSFORMATION MATRIX R)
      CALL GAUSSI (A,B,NRET,KR)
C   B = T(INV)
C   TRANSFORM STATE VECTOR FOR POSSIBLE USE IN LINEARIZED RESPONSE.
      K = 0
      DO 15 I=1,NEQ
      IF (INDEF(I) .EQ. 0) GO TO 15
      K=K+1
      Y(K) = Y(I)
   15 CONTINUE
C          FIND Z VECTORS IN EQUATION III-23 VOL 1
      CALL MULTB (B,Y,NRET,NRET,1,KR,KR)
      CALL WRITE (Y,1,NRET,4H Y* ,1)
C          READ PARTIAL DERIVITIVE MATRIX H
      READ (NUT2) ((A(I,J),I=1,KR),J=1,KR)
C          FORM R**(-1)*H
      CALL MULTA (B,A,NX,NX,NX,KR,KR)
C   B= T(INV) * -A-
C          READ H
```

```
      READ (NOT2) ((A(I,J),I=1,KR),J=1,KR)                               0 7326
      REWIND NOT2                                                        C 7327
C          FORM R**(-1)*H*R EQUATION III-24 PUT RESULTS IN A ARRAY       0 7328
      CALL MULTB (B,A,NX,NX,NX,KR,KR)                                    0 7329
      IF(.NOT.LEQU) GO TO 150                                            0 7330
      WRITE (NOT,1006)                                                   0 7331
 1006 FORMAT (' R**(-1)*H*R MATRIX IS ')                                 C 7332
      CALL WRITES (A,NX,NX,KR)                                           0 7333
  150 CONTINUE                                                           C 7334
C B = -A*-                                                                0 7335
C                                                                        C 7336
C  REORDER    FROM Y,DELTA,XSS,B                                         0 7337
C                  TO Y,XSS,DELTA,B                                      0 7338
C          NY2 = NX - NDLTA - NXSS                                       C 7339
C          ND2 = NDLTA - NBTQ                                            0 7340
C                                                                        0 7341
      DO 20 I=1,NX                                                       0 7342
   20 IV(I) = I                                                          C 7343
      DO 30 I=1,NXSS                                                     0 7344
      L = NY2 + I                                                        C 7345
      K = L + ND2                                                        C 7346
   30 IV(K) = L                                                          0 7347
      IF(ND2.EQ.0) GO TO 41                                              C 7348
      DO 40 I=1,ND2                                                      0 7349
      K = NY2 + I                                                        0 7350
      L = NY2 + NXSS + I                                                 C 7351
   40 IV(K) = L                                                          0 7352
   41 CONTINUE                                                           0 7353
      CALL ZERO (B,NX,NX,KR)                                             0 7354
      CALL REVADD (1.D0,A,IV,IV,B,NX,NX,NX,NX,KR,KR)                     0 7355
      IF(.NOT.LEQU) GO TO 160                                            C 7356
      WRITE(NOT,1007)                                                    0 7357
 1007 FORMAT (' REORDERED R**(-1)*H*R MATRIX IS ')                       0 7358
      CALL WRITES(B,NX,NX,KR)                                            0 7359
  160 CONTINUE                                                           0 7360
C                                                                        0 7361
      RETURN                                                             C 7362
      END                                                                0 7363
```

```
      SUBROUTINE FINDT (C,NCN,NX,NS,T,NRET,KC,KT)                        0 7364
C                                                        DEBUG #  66     0 7365
      IMPLICIT REAL*8(A-H,O-Z)                                           0 7366
C                                                                        0 7367
C     ---LATEST MOD ON 09/04/75                                          C 7368
C                                                                        0 7369
      DIMENSION C(KC,1), T(KT,1)                                         0 7370
      DIMENSION IVEC(207), JVEC(207)                                   448 7371
C                                                                        0 7372
      COMMON /LQSIZE/                                                    0 7373
     2              NNX, NY, NDLTA, NXSS, NBTQ, NJQ, NY2, ND2            0 7374
           COMMON /SPECIF/                                               0 7375
     *    BETAH(6, 6),BETAHD(6, 6),AMD(2, 5),RH(3,3,30),RS(3,3,30),     16 7376
     *    DH(3,35),DS(3,30),IMU(5, 5),NMUW(6, 6),IFTSAW(15),            17 7377
     *    NB,NH,NSPT,NOFMC,NDELTA,ITCPOL(2, 6),IRGFLX( 6),IHDATA(7, 6), 18 7378
     *    LOCC(14),LENG(14),NO,NDETA,NLAM,NEQ                           19 7379
```

```
      COMMON /LIJV  /                                                   0 7380
     9              IV(250)  , JV  (250)                              420 7381
                COMMON /VINDEP/                                         0 7382
     *      INDEP(250)                                                 21 7383
      LOGICAL DEBUG,LEQU                                                0 7384
      COMMON /LDEBUG/ DEBUG(120)                                        0 7385
      EQUIVALENCE (LEQU,DEBUG(66))                                      0 7386
C                                                                       0 7387
      DATA EPS , NOT / 1.D-15, 6 /                                      0 7388
 1001 FORMAT (/// 5X,36HSUBROUTINE FIND TERMINATED, SING. AT   I5)      0 7389
 1002 FORMAT (///1X,                                                    0 7390
     * 55HTHE TRANSFORMED STATE VECTOR CORRELATION ARRAY FOLLOWS. /1X,  0 7391
     * 43HELEMENTS REFER TO ORIGINAL STATE VARIABLES.,/1X,115(1H-))     0 7392
 2001 FORMAT (////1X,47HTHE FOLLOWING INTEGER ARRAY (INDEP) PRESCRIBES  0 7393
     *54HINDEPENDENT VARIABLES (1), AND DEPENDENT VARIABLES (0). /1X,   0 7394
     * 115(1H-))                                                        0 7395
 2010 FORMAT (//1X44HTHE STATE VECTOR LENGTH ARRAY (LENQ) FOLLOWS ,     0 7396
     *     /1X,115(1H-))                                                0 7397
 2011 FORMAT (//1X46HTHE STATE VECTOR LOCATION ARRAY (LOCQ) FOLLOWS.    0 7398
     *     /1X,115(1H-))                                                0 7399
C                                                                       0 7400
      IF(.NOT.LEQU) GO TO 120                                           0 7401
      WRITE(NOT,1003)                                                   0 7402
 1003 FORMAT (' SUBROUTINE FINDT INPUT MATRIX ')                        0 7403
      CALL WRITES (C,NCN,NX,KC)                                         0 7404
  120 CONTINUE                                                          0 7405
      DO 5 I=1,NX                                                       0 7406
    5 JVEC(I) = 1                                                       0 7407
C                                                                       0 7408
C         FOR PROCEDURE SEE VOL 1,PAGE 63 'ILLUSTRATIVE EXAMPLE'        0 7409
      DO 10 L=1,NCN                                                     0 7410
      JBIG = 1                                                          0 7411
      A = DABS(C(L,1))                                                  0 7412
C         FIND PIVOT ELEMENT FOR ROW L                                  0 7413
      DO 15 J=2,NX                                                      0 7414
      AT = DABS(C(L,J))                                                 0 7415
      IF (AT .LT. A) GO TO 15                                           0 7416
      A = AT                                                            0 7417
      JBIG = J                                                          0 7418
   15 CONTINUE                                                          0 7419
C         PIVOT ELEMENT FOR ROW L IS C(L,JBIG)                          0 7420
      IVEC(L) = -JBIG                                                   0 7421
      JVEC(JBIG) = 0                                                    0 7422
      IF (A .GT. EPS) GO TO 20                                          0 7423
      WRITE (NOT,1001) L                                                0 7424
      STOP                                                              0 7425
   20 CONTINUE                                                          0 7426
      CLJBIG = C(L,JBIG)                                                0 7427
      DO 17 J=1,NX                                                      0 7428
   17 C(L,J) = C(L,J)/CLJBIG                                            0 7429
      DO 25 I=1,NCN                                                     0 7430
      F = C(I,JBIG)                                                     0 7431
      IF (I .EQ. L) GO TO 25                                            0 7432
      DO 30 J=1,NX                                                      0 7433
   30 C(I,J) = C(I,J) - F*C(L,J)                                        0 7434
   25 CONTINUE                                                          0 7435
   10 CONTINUE                                                          0 7436
C         LOWER CASE D SUB IJ MATRIX ON PAGE 64 FOUND)                  0 7437
      IF(.NOT.LEQU) GO TO 130                                           0 7438
      WRITE(NOT,1004)                                                   0 7439
 1020 FORMAT (' VARIABLE EXCHANGE MATRIX IS ')
      CALL WRITM(E,NX,NX,NC)
  120 CONTINUE
C
      NVAL = 0
      DO 40 I=1,NX
      IF (JVEC(I) .EQ. 0) GO TO 40
      NVAL = NVAL + 1
      JVEC(I) = NVAL
   40 CONTINUE
C
C          CONSTRUCT THE NEW SIMILARITY TRANSFORMATION MATRIX
C          SEE EXAMPLE PAGE C4
      NSEL = NX - NCN
      CALL ZERO    (T,NSEL,NSEL,KT)
      CALL [illegible] (I,00,[illegible],JVEC,T,NCN,NX,NRC,[illegible],ISEL,KT)
C
      DO 50 I=1,NX
      IF (JVEC(I) .EQ. 0) GO TO 50
      NK = JVEC(I)
      T(I,NK) = 1.00
   50 CONTINUE
C
      IF([illegible].LE.00) GO TO 140
      WRITE(NOT,1030)
 1030 FORMAT (' SIMILARITY TRANSFORMATION MATRIX IS ')
      CALL WRITM(T,NSEL,NSEL,KT)
  140 CONTINUE
C
      NAUX = NASS + NOTG
      NSV = NEQ + NAUX
C
C        THE [illegible] EQUATION VECTOR IDENTIFIES Y* VARIABLES.
C
      K = 0
      DO 60 I=1,NSV
      IF (INDEP(I) .EQ. 0) GO TO 60
      K=K+1
      IV(K) = I
   60 CONTINUE
C
      K=0
      DO 70 I=1,NX
      IF (JVEC(I) .EQ. 0) GO TO 70
      K=K+1
      JV(K) = IV(I)
   70 CONTINUE
C
      NZ = NY2 + ND2 + NASS + NOTG
C
      DO 90 I=1,NZ
   90 IV(I) = JV(I)
C
      DO 100 I=1,NASS
      L=NY2+I
      K=L+ND2
  100 IV(L) = JV(K)
C
      IF(ND2.EQ.0) GO TO 111
      DO 110 I=1,ND2
      K = NY2 + 1
      L = K + NXSS
  110 IV(L) = JV(K)
  111 CONTINUE
C
      CALL PAGEHD
      LEQU = 2*ND + 2
      NEQP = NEQ + NAUX
      WRITE (NOT,2001)
      CALL WRITIS (INDEP,1,NEQP,1)
      WRITE (NOT,2010)
      CALL WRITIS (LENU ,1,LEQU,1)
      WRITE (NOT,2011)
      CALL WRITIS (LOCU ,1,LEQU,1)
      WRITE (NOT,1002)
      CALL WRITIS (IV,1,NZ,1)
C
C
      RETURN
      END
```

```
      SUBROUTINE TFTYPE (A,Z,B,NA,NZ,ITYPE,JCOL,NOKP,KOKP,KA,KZ)
C
C                                                        DEBUG #  67
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /LDSIZE/
     2                 NX, NY, NDLTA, NXSS, NB, NJJ, NY2, NJ2
      LOGICAL DEBUG,LEQU
      COMMON /LDDBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(67))
      DATA NOT/6/
C
      DIMENSION A(KA,1), Z(KZ,1), B(1), KOKP(1)
C
C         ----- SUBROUTINE ARGUMENT DESCRIPTIONS -----
C
C   A       = INPUT   PARTIAL DERIVATIVE MATRIX
C   Z       = OUTPUT  REDUCED PARTIAL DERIVATIVE MATRIX.  (NZ,NZ)
C   B       = OUTPUT  VECTOR OF COEFF. FOR DESIRED TF INPUT. (NZ,1)
C   NA      =  INPUT  SIZE OF A.
C   NZ      = OUTPUT  SIZE OF Z
C   ITYPE   = INPUT  =1  FORWARD PATH TF    XSS(I)/RT(J)
C                     2  FEEDBACK TF        B(I)/XSS(J)
C                     3  OPEN LOOP TF       B(I)/RT(J)
C                     4  OPEN LOOP TF       XSS(I)/RS(J)
C                     5 CLOSED LOOP TF      XSS(I)/RT(J)
C                     6 CLOSED LOOP TF      XSS(I)/RS(J)
C                     7 PARTIAL OPEN LOOP  B(I)/RT(J)
C                     8 SPECIAL CASE SEE BELOW
C
C            NOTE-- A MINUS SIGN ON ITYPE INDICATES
C                   NEGATIVE FEEDBACK FOR NUMERATOR
C                   AUGMENTATION SELECTION OF PROPER B.
C
```

```
C  JCOL    = INPUT  COL LOCATION IN A OF DESIRED INPUT(J). LOCAL
C        FOR SENSOR SIGNALS
C             0 < JCOL < NXSS+1
C        FOR TORQUE SIGNALS
C             0 < JCOL < NU+1
C  NUKP    = INPUT NO. OF B'S TO RETAIN  ITYPE=7
C  KEKP    = INPUT ID VECTOR NOTING WHICH B'S TO KEEP (LOCAL)
C  KA      = INPUT  ROW DIMENSION OF  A IN CALLING PROGRAM
C  KZ      = INPUT  ROW DIMENSION OF  Z IN CALLING PROGRAM
C
C        ESTABLISH LEADING ELE LOCATORS FOR EACH PARTITION OF A
C        ASSUMED ORDER IS Y,XSS,DELTA,B
C
C        LY = LEADING ELEMENT FOR STATE VARIABLES
C        LX - LEADING ELEMENT FOR SENSOR SIGNALS
C        LU - LEADING ELEMENT FOR CONTROL STATE VARIABLES
C        LE = LEADING ELEMENT FOR TORQUE SIGNALS
C        KSN = +1,-1, POSITIVE, NEGITIVE FEEDBACK
C
      LY = 1
      LX = LY + NY2
      LU = LX + NXSS
      LE = LU + NU2
      KSN = ISIGN(1,ITYPE)
      ITYPE - IABS(ITYPE)
C
                      NERROR = 1
      IF (ITYPE .LT. 1  .OR.  ITYPE .GT. 8) GO TO 999
                      NERROR = 2
      IF (JCOL .LT. 1  .OR.  JCOL .GT. NA) GO TO 999
C
      GO TO (1,2,3,4,5,6,7,8), ITYPE
C
    1 CONTINUE
C               -----ITYPE = 1 -----
C        TYPE 1 FORWARD PATH TRANSFER FUNCTION FOR PLANT WITH NO
C                FEEDBACK, EQUATION III-49, VOL 1
C
C      PLANT Z = A11,A12
C                A21,A22
C
C            E = A1+(JCOL)*KSN
C                A2+(JCOL)*KSN
C
      NZ = NY2 + NXSS
C        KCOL = ABSOLUTE LOCATION OF  TORQUE SIGNAL
      KCOL - LE-1+JCOL
      DO 10 I=1,NZ
      E(I) = A(I,KCOL) * KSN
      DO 10 J=1,NZ
   10 Z(I,J) = A(I,J)
      IF(.NOT.LOGO) RETURN
      WRITE (NOT,1000) ITYPE
      WRITE (NOT,1001)
      CALL WRITEM(Z,NZ,NZ,KZ)
      WRITE (NOT,1002)
      CALL WRITEM(E,1,NZ,1)
      RETURN
C
    2 CONTINUE
```

```
C              -----ITYPE = 2 -----
C         TYPE 2 FEEDBCK PATH H(S) FOR THE CONTROLLER ONLY. THE TRANSFER
C                FUNCTION RELATES CONTROL SYSTEM OUTPUT TO SENSOR SIGNAL
C                INPUT EQUATION III-53, VOL 1
C
C       FORM Z = A33,A34
C                A43,A44
C
C            B = A32(JCOL)
C                A42(JCOL)
C
      NZ = NO2 + NB
C        KCOL = ABSOLUTE LOCATION OF  SENSOR SIGNAL
      KCOL = LX-1+JCOL
      CALL ZERO (Z,NZ,NZ,KZ)
      DO 20 I=1,NZ
      IRA = I + NY2 + NXSS
      B(I) = A(IRA,KCOL)
      DO 20 J=1,NZ
      JCA = J + NY2 + NXSS
      Z(I,J) = A(IRA,JCA)
   20 CONTINUE
      IF(.NOT.LEQU) RETURN
      WRITE (NOT,1000) ITYPE
      WRITE (NOT,1001)
      CALL WRITES(Z,NZ,NZ,KZ)
      WRITE (NOT,1002)
      CALL WRITES(B,1,NZ,1)
      RETURN
C
    3 CONTINUE
C              -----ITYPE = 3 -----
C         TYPE 3 CLASSICAL OPEN LOOP TRANSFER FUNCTION RELATES CONTROL
C                SYSTEM OUTPUT TO EXTERNAL PLANT INPUT. EQUATION III-55
C
C       FORM Z = A11,A12,A13,  0
C                A21,A22,A23,  0
C                A31,A32,A33,A34
C                A41,A42,A43,A44
C
C            B = A14(JCOL)*XSN
C                A24(JCOL)*XSN
C                  0
C                  0
C
      NZ = NY2 + NXSS + NO2 + NB
C        KCOL = ABSOLUTE LOCATION OF  TORQUE SIGNAL
      KCOL = LE-1+JCOL
      CALL ZERO (Z,NZ,NZ,KZ)
      M = NY2 + NXSS
      DO 25 I=1,NZ
      B(I) = A(I,KCOL)*XSN
      DO 25 J=1,NZ
   25 Z(I,J) = A(I,J)
      DO 26 I=1,M
      DO 26 J=LB,NZ
   26 Z(I,J) = 0.D0
      DO 30 I=LB,NZ
   30 B(I) = 0.D0
      IF(.NOT.LEQU) RETURN
      WRITE (NCT,1000) ITYPE                                            0 7674
      WRITE (NCT,1001)                                                  0 7675
      CALL WRITES(Z,NZ,NZ,KZ)                                           0 7676
      WRITE (NCT,1002)                                                  0 7677
      CALL WRITES(B,1,NZ,1)                                             0 7678
      RETURN                                                            0 7679
C                                                                       0 7680
    4 CONTINUE                                                          0 7681
C                -----ITYPE = 4 -----                                   0 7682
C        TYPE 4 OPEN LOOP TRANSFER FUNCTION, RELATES PLANT SENSOR SIGNAL 0 7683
C               OUTPUT TO CONTROLLER NOISE INPUT, EQUATION III-57       0 7684
C                                                                       0 7685
C       FORM Z = A11,A12,A13,A14                                        0 7686
C                A21,A22,A23,A24                                        0 7687
C                A31,  0,A33,A34                                        0 7688
C                A41,  0,A43,A44                                        0 7689
C                                                                       0 7690
C            B =   0                                                    0 7691
C                  0                                                    0 7692
C                A32(JCOL)                                              0 7693
C                A42(JCOL)                                              0 7694
C                                                                       0 7695
      NZ = NY2 + NXSS + NU2 + NB                                        0 7696
C        KCOL = ABSOLUTE LOCATION OF SENSOR SIGNAL                      0 7697
      KCOL = LX-1+JCOL                                                  0 7698
      CALL ZERO (Z,NZ,NZ,KZ)                                            0 7699
      M = NY2 + NXSS                                                    0 7700
      DO 35 I=1,NZ                                                      0 7701
      B(I) = A(I,KCOL)                                                  0 7702
      DO 35 J=1,NZ                                                      0 7703
   35 Z(I,J) = A(I,J)                                                   0 7704
      DO 36 I=LU,NZ                                                     0 7705
      DO 36 J=LX,M                                                      0 7706
   36 Z(I,J) = 0.00                                                     0 7707
      DO 40 I=1,M                                                       0 7708
   40 B(I) = 0.00                                                       0 7709
      IF(.NOT.LEQUC) RETURN                                             0 7710
      WRITE (NCT,1000) ITYPE                                            0 7711
      WRITE (NCT,1001)                                                  0 7712
      CALL WRITES(Z,NZ,NZ,KZ)                                           0 7713
      WRITE (NCT,1002)                                                  0 7714
      CALL WRITES(B,1,NZ,1)                                             0 7715
      RETURN                                                            0 7716
C                                                                       0 7717
    5 CONTINUE                                                          0 7718
C                -----ITYPE = 5 -----                                   0 7719
C        TYPE 5 TRANSFER FUNCTION THAT RELATES SENSOR SIGNAL OUTPUT     0 7720
C               TO EXTERNALLY APPLIED PLANT INPUTS WITH THE CONTROL     0 7721
C               SYSTEM ENTIRELY ACTIVE, EQUATION III-59                 0 7722
C                                                                       0 7723
C       FORM Z = A11,A12,A13,A14                                        0 7724
C                A21,A22,A23,A24                                        0 7725
C                A31,A32,A33,A34                                        0 7726
C                A41,A42,A43,A44                                        0 7727
C                                                                       0 7728
C            B = A14(JCOL)*XSN                                          0 7729
C                A24(JCOL)*XSN                                          0 7730
C                  0                                                    0 7731
C                  0                                                    0 7732
      NZ = NY2 + NXSS + NU2 + NB                                        0 7733
```

```
C         KCOL = ABSOLUTE LOCATION OF TORQUE SIGNAL
      KCOL = LB-1+JCOL
      CALL ZERO (Z,NZ,NZ,KZ)
      DO 45 I=1,NZ
      B(I) = A(I,KCOL)*XSN
      DO 45 J=1,NZ
   45 Z(I,J) = A(I,J)
      DO 50 I=LD,NZ
   50 B(I) = 0.D0
      IF(.NOT.LEQU) RETURN
      WRITE (NCT,1000) ITYPE
      WRITE (NCT,1001)
      CALL WRITES(Z,NZ,NZ,KZ)
      WRITE (NCT,1002)
      CALL WRITES(B,1,NZ,1)
      RETURN
C
    6 CONTINUE
C                -----ITYPE = 6 -----
C         TYPE 6 CLOSED LOOP TRANSFER FUNCTION RELATES PLANT SENSOR
C                SIGNAL OUTPUT TO SENSOR SIGNAL NOISE INPUT ALL CONTROL
C                SYSTEM LOOPS ACTIVE, EQUATION III-61
C
C        FORM Z = A11,A12,A13,A14
C                 A21,A22,A23,A24
C                 A31,A32,A33,A34
C                 A41,A42,A43,A44
C
C            B =  0
C                 0
C                 A32(JCOL)
C                 A42(JCOL)
C
      NZ = NY2 + NXSS + NU2 + NB
C         KCOL = ABSOLUTE LOCATION OF SENSOR SIGNAL
      KCOL = LX-1+JCOL
      CALL ZERO (Z,NZ,NZ,KZ)
      M = NY2 + NXSS
      DO 55 I=1,NZ
      B(I) = A(I,KCOL)
      DO 55 J=1,NZ
   55 Z(I,J) = A(I,J)
      DO 60 I=1,M
   60 B(I) = 0.D0
      IF(.NOT.LEQU) RETURN
      WRITE (NCT,1000) ITYPE
      WRITE (NCT,1001)
      CALL WRITES(Z,NZ,NZ,KZ)
      WRITE (NCT,1002)
      CALL WRITES(B,1,NZ,1)
      RETURN
C
C
    7 CONTINUE
C                -----ITYPE = 7-----
C         TYPE 7 QUASI OPEN LOOP TRANSFER FUNCTION RELATES CONTROL SYSTEM
C                OUTPUTS DUE TO PLANT VARIABLE INPUTS WITH THE OPTION TO
C                OPEN SOME CONTROLLER OUTPUT CHANNELS. MAX NUMBER OF
C                FEEDBACK SIGNALS IS 3. SEE PAGE 77,VOL I
C                   NBKP = NUMBER OF FEEDBACK SIGNALS
```

```
C                    KBKP(I) = LOCAL LOCATION OF SIGNAL TO FEEDBACK
C
C        FORM Z = A11,A12,A13,(A14)
C                 A21,A22,A23,(A24)
C                 A31,A32,A33,A34
C                 A41,A42,A43,A44
C
C             B = A14(JCOL)*XSN
C                 A24(JCOL)*XSN
C                  0
C                  0
C
      NZ = NY2 + NXSS + ND2 + NB
C         KCOL = ABSOLUTE LOCATION OF TORQUE SIGNAL
      KCOL = LB-1+JCOL
      CALL ZERO (Z,NZ,NZ,KZ)
      M = NY2 + NXSS
      M1 = M + ND2
      DO 65 I=1,M
      B(I) = A(I,KCOL) * XSN
      DO 62 J=1,M1
   62 Z(I,J) = A(I,J)
      DO 63 J=1,NBKP
C         LCOL = ABSOLUTE LOCATION OF TORQUE SIGNAL TO FEEDBACK
C                0 < KBKP(J) < NB+1
      LCOL = LC-1+KBKP(J)
   63 Z(I,LCOL) = A(I,LCOL)
   65 CONTINUE
      DO 70 I=LD,NZ
      B(I) = 0.D0
      DO 70 J=1,NZ
      Z(I,J) = A(I,J)
   70 CONTINUE
      IF (.NOT.LPGO) RETURN
      WRITE (NOT,1000) ITYPE
      WRITE (NOT,1001)
      CALL WRITES(Z,NZ,NZ,KZ)
      WRITE (NOT,1002)
      CALL WRITES(B,1,NZ,1)
      RETURN
C
C
    8 CONTINUE
C              -----IABS(ITYPE) = 8 -----
C           ITYPE 8 CLOSED LOOP TRANSFER FUNCTION BETWEEN ANY SYSTEM STATE
C                 VARIABLE (JCOL NOT LOCAL) OUTPUT TO ANY SYSTEM STATE
C                 VARIABLE NOISE INPUT
C                    ITYPE =  8 FEEDBACK LOOP CUT
C                    ITYPE = -8 CLOSED LOOP
C
C        FORM FOR ITYPE = -8
C
C                 A11,A12.....A1J......A1N          -A1J
C                 A21,A22.....A2J......A2N          -A2J
C                  .   .       .        .              .
C                  .   .       .        .              .
C                  .   .       .        .              .
C        Z =      AJ1,AJ2.....AJJ......AJN      B =    0
C                  .   .       .        .              .
C                  .   .       .        .              .
```

```
C                 .   .        .       .          .
C              AN1,AN2,....,ANJ,....,ANN          -ANJ
C
C       WHERE:
C              J = JCOL
C              N = NZ
C              ELEMENTS OF Z EQUATED TO A MATRIX
C              ELEMENTS OF B EQUATED TO NEGATIVE OF COLUMN JCOL
C                        OF A WITH ROW JCOL OF B SET EQUAL TO ZERO
C
C       FORM FOR ITYPE =  8
C
C              A11,A12,...., 0 .....,A1N          -A1J
C              A21,A22,...., 0 .....,A2N          -A2J
C               .   .        .       .             .
C               .   .        .       .             .
C               .   .        .       .             .
C       Z =    AJ1,AJ2,....,AJJ,....,AJN      B =    0
C               .   .        .       .             .
C               .   .        .       .             .
C               .   .        .       .             .
C              AN1,AN2,...., 0 .....,ANN          -ANJ
C
C       WHERE:
C              J = JCOL
C              N = NZ
C              ELEMENTS OF Z EQUATED TO A MATRIX, COLUMN JCOL RESET
C                       TO ZERO, (JCOL,JCOL) ELEMENT OF Z
C                        RESET AGAIN TO A(JCOL,JCOL)
C              ELEMENTS OF B SAME AS IN ITYPE = -8
C
CCC   FOR ITYPE = 8, LRY(2,ICYC) IS GLOBAL REFERENCE INPUT STATE OF Y*,
CCC                               ((WHOSE FEEDBACK LOOP IS CUT)),
CCC                   LRY(3,ICYC) IS GLOBAL OUTPUT STATE OF Y*
C
CCC   FOR ITYPE = -8, LRY(2,ICYC) AND LRY(3,ICYC) ARE THE SAME
CCC                   (AS ITYPE = 8), BUT THIS IS A COMPLETELY CLOSED
CCC                   LOOP TRANSFER FUNCTION.
C
      NZ = NY2 + NXSS + NU2 + NB
      DO 75 I=1,NZ
      B(I) = -A(I,JCOL)
      DO 75 J=1,NZ
   75 Z(I,J) = A(I,J)
      IF (XSN .LT. 0.D0) GO TO 77
      DO 76 I=1,NZ
   76 Z(I,JCOL) = 0.D0
   77 Z(JCOL,JCOL) = -B(JCOL)
      B(JCOL) = 0.D0
      IF(.NOT.LEQU) RETURN
      WRITE (NOT,1000) ITYPE
      WRITE (NOT,1001)
      CALL WRITES(Z,NZ,NZ,KZ)
      WRITE (NOT,1002)
      CALL WRITES(B,1,NZ,1)
      RETURN
C
C
  999 CONTINUE
      WRITE (6,2001) NERROR
```

```
C
 1000 FORMAT (' SUBROUTINE TFTYPE, TRANSFER FUNCTION TYPE',I3)
 1001 FORMAT (' THE CHARACTERISTIC MATRIX IS ')
 1002 FORMAT (' THE CORRESPONDING INPUT COEFFICIENTS ARE ')
 2001 FORMAT (1H1,5X,40HPROGRAM STOPPED IN SUBROUTINE TFTYPE.  NERROR =
     *    , I3)
      STOP
      END




      SUBROUTINE NUMS (A,D,B,R1R,R1I,R2R,R2I,PTOL,
     *                 GAIN,IFLG,NN,NZRO,IY,NA,KA)
C
C                                                          DEBUG #  68
      IMPLICIT REAL*8(A-H,O-Z)
C
C     ---LATEST MOD ON 09/04/75
C
C   SUBROUTINE DETERMINES NUM(S)
C        THAT IS THE NUMERATOR OF TRANSFER FUNCTION SET UP THROUGH
C         CALLS FROM DYNS40
C   CALLS OR PACKAGE
C
      COMMON /ORATIO/
     *      IFL1,IFL2,DRVEC(150)
      LOGICAL DEBUG,LEQU
      COMMON /DEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(68))
C
      DIMENSION A(KA,1),D(1),R1R(1),R1I(1),R2R(1),R2I(1)
      DIMENSION B(KA,1)
C
C               -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C   A       = INPUT  TRANSFORMED PARTIAL DERIVATIVE MATRIX.  NA,NA
C   B       = INPUT  COEFF. OF INPUT TF FOR Q(OUT)/Q(IN).  NA,1
C   R1R     = OUTPUT REAL ROOTS OF FIRST TERM IN NUMERATOR.  NA-1
C   R1I     = OUTPUT IMAGINARY ROOTS OF FIRST TERM IN NUMERATOR.
C   R2R     = OUTPUT  REAL ROOTS OF SECOND TERM IN NUMERATOR.  NA-1
C   R2I     = IMAGINARY ROOTS OF SECOND TERM IN NUMERATOR.
C             FOR INVERSE SHIFT.
C   PTOL    = INPUT TOLERANCE TO REMOVE THE P(I) = 0 ROOTS
C   GAIN    = OUTPUT GAIN OF NUMERATOR.
C   NN      = OUTPUT NUMBER OF NUMERATOR TERMS.  EITHER 1 OR 2.
C   IY      = INPUT  COL LOCATION OF DESIRED Q(OUT) --ABSOLUTE--
C   KA      = INPUT  ROW DIMENSION SIZE OF A IN CALLING PROGRAM.
C   IFLG    = FLAG TO SET FOR ZERO GAIN.
C           = 0 ZERO GAIN DETECTED.
C           =1 GAIN NOT ZERO.
C
      CON = -DSQRT(3.D0)
C
      IFL1 = 1
      IFL2 = 0
      IFLG = 1
C
```

```
C     FORM AUGMENTED A MATRIX
C     REPLACE COL IY OF A WITH COL NA OF A AND
C     PUT B INTO COL NA OF A
C
      DO 10 I=1,NA
      A(I,IY) = A(I,NA)
   10 A(I,NA) = -B(I)
C      INTERCHANGE ROW IY OF A WITH ROW NA OF A
      DO 15 J=1,NA
      Z = A(IY,J)
      A(IY,J) = A(NA,J)
   15 A(NA,J) = Z
C      OBTAIN PIVOTAL ELE AND CK FOR ZERO
      Z = A(NA,NA)
      IF (Z .EQ. 0.D0) GO TO 50
      GAIN = -Z
      NN = 1
      DO 20 J=1,NA
   20 A(NA,J) = A(NA,J) / Z
      NA2 = NA-1
      DO 30 I=1,NA2
      DO 30 J=1,NA
   30 A(I,J) = A(I,J) - A(I,NA) * A(NA,J)
      CALL WRITE (A,NA,NA,4HANUM,KA)
C         GET ROOTS OF AUGMENTED A MATRIX IF ALL PIVOT ELEMENTS NON-ZERO
      CALL GRDRVR (A,NA2,R1R,R1I,KA)
      NZRO = NA2
      GO TO 100
C         COME HERE IF PIVOT ELEMENT ZERO,SEE BOTTOM PAGE 71,VOL 1
   50 CONTINUE
C
      NN = 1
      NA2 = NA-1
C
C     FORM (A - I*CON)
C
      DO 55 I=1,NA2
   55 A(I,I) = A(I,I) - CON
C
C         CALL GAUSSI TO GET  (A - I*CON)**(-1) RETURN IT IN D ARRAY
      CALL GAUSSI (A,D,NA,KA)
      IF (IFL2 .EQ. 0) GO TO 57
      IFL2 = 0
      IFLG = 0
C
C     ZERO GAIN ENCOUNTERED IN GAUSSI.
C
      RETURN
C
   57 CONTINUE
C
C         FORM PRODUCT DEFINED BY EQUATION III-41
C              NOTE THE I TILDA MATRIX HAS ZERO IN NA,NA POSITION
      DO 59 I=1,NA
   59 D(I,NA) = 0.D0
C
C         FIND ROOTS OF EQUATION III-42
      IF(LEQU) CALL WRITE(D,NA,NA,6H -D-   ,KA)
      CALL GRDRVR (D,NA,R2R,R2I,KA)
C
```

```
C     REMOVE THE P(I) = 0. ROOTS.                                         0 8028
C                                                                         0 8029
      CALL SIFT (R2R,NA,PTOL)                                             0 8030
      CALL SIFT (R2I,NA,PTOL)                                             0 8031
C                                                                         0 8032
      K=0                                                                 0 8033
      DO 60 I=1,NA                                                        0 8034
      IF (R2R(I) .EQ. 0.D0  .AND.                                         0 8035
     *    R2I(I) .EQ. 0.D0)        GO TO 60                               0 8036
      K=K+1                                                               0 8037
      R1R(K) = R2R(I)                                                     0 8038
      R1I(K) = R2I(I)                                                     0 8039
   60 CONTINUE                                                            0 8040
C                                                                         0 8041
C         GET NUMERATOR GAIN SEE EQUATION III-48                          0 8042
C         NZRO = NUMBER OF NON-ZERO ROOTS IN NUMERATOR EQUATION           0 8043
      NZRO = K                                                            0 8044
      DO 70 I=1,NA                                                        0 8045
      IF (I .GT. NZRO) GO TO 71                                           0 8046
      IF (R1I(I) .EQ. 0.D0) GO TO 65                                      0 8047
      DRVEC(I) = DRVEC(I)*DSQRT(R1R(I)**2 + R1I(I)**2)                    0 8048
      GO TO 70                                                            0 8049
   65 DRVEC(I) = DRVEC(I)*R1R(I)                                          0 8050
   70 CONTINUE                                                            0 8051
C         FORM PRODUCT DEFINED BY EQUATION III-48 TO OBTAIN               0 8052
C                NUMERATOR GAIN                                           0 8053
   71 GAIN = DRVEC(1)                                                     0 8054
      DO 58 I=2,NA                                                        0 8055
   58 GAIN = GAIN*DRVEC(I)                                                0 8056
      IF (MOD((NA-NZRO),2) .EQ. 1) GAIN = -GAIN                           0 8057
C                                                                         0 8058
C     REMOVE THE SHIFT VALUE TO OBTAIN TRUE ROOTS.                        0 8059
C              S(I) = 1.0/P(I) + CON     S AND P COMPLEX OF COURSE        0 8060
C                                                                         0 8061
      DO 80 I=1,NZRO                                                      0 8062
      RMOD = R1R(I)**2 + R1I(I)**2                                        0 8063
      R1R(I) = R1R(I)/RMOD + CON                                          0 8064
   80 R1I(I) = -R1I(I)/RMOD                                               0 8065
C                                                                         0 8066
      CALL WRITE (DRVEC,1,NA,6HDRATIO,1)                                  0 8067
  100 CONTINUE                                                            0 8068
      IF(LEGO) CALL WRITE(R1R,1,NZRO,6H -R1R-,1)                          0 8069
      IF(LEGO) CALL WRITE(R1I,1,NZRO,6H -R1I-,1)                          0 8070
      IF(LEGO) CALL WRITE(GAIN,1,1,6H GAIN ,1)                            0 8071
      RETURN                                                              0 8072
      END                                                                 0 8073
```

```
      SUBROUTINE TFPLY (A,B,C,K,NS,L)                                     0 8074
C                                                          DEBUG #  69    0 8075
      IMPLICIT REAL*8 (A-H,O-Z)                                           0 8076
C                                                                         0 8077
      DIMENSION A(1), B(1)                                                0 8078
C                                                                         0 8079
C      SUBROUTINE CONVERTS TRANSFER FUNCTION POLYNOMINAL                  0 8080
C                 EXPRESSIONS TO FIRST ORDER CANONICAL STATE              0 8081
```

```
C                    SPACE FORM AND RETURNS THE TRANSFORMED OUTPUT
C                     VARIABLE, X.
C
C                                          ----------
C                                          I        I
C                              U           I        I     X
C                           ------- >  I  TF(S)   I ------ >
C                                          I        I
C                                          I        I
C                                          ----------
C
C                   B(1) + B(2)*S + B(3)*S**2 + ... + B(KS)*S**(KS-1)
C        TF(S) =   ---------------------------------------------------
C                   A(1) + A(2)*S + A(3)*S**2 + ... + A(NS)*S**(NS-1)
C
C         WHERE KS.LE.NS
C
C
C                -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C   A        = INPUT   VECTOR OF DENOMINATOR POLYNOMINAL
C                      COEFFICIENTS--ASCENDING ORDER.
C   B        = INPUT   VECTOR OF NUMERATOR POLYNOMINAL
C                      COEFFICIENTS--ASCENDING ORDER.
C   U        = INPUT   STATE VARIABLE TO BE OPERATED ON BY THE
C                      POLYNOMINAL TRANSFER FUNCTION.
C   X        = OUTPUT  VARIABLE RESULTING FROM THE TRANSFER
C                      FUNCTION OPERATING ON U.
C   NS       = INPUT   SIZE OF A AND B.
C   L        = INPUT   LOCATION (IN STATE VECTOR) OF THE
C                      LEADING ELEMENT OF THE NS-1 STATE VARIABLES
C                      ESTABLISHED FROM THE POLYNOMINALS.
C
C         FOR THEORY SEE VOL I,PAGE 78
C
      COMMON /VECTOR/
     E                  Y    (250), YD   (250)
      DATA NIT/ 6 /
C
C   NORMALIZE A AND B COEFFICIENTS TO COEFFICIENT OF
C   HIGHEST DERIVATIVE IN DENOMINATOR, A(NS).
C
      AN = A(NS)
      IF (AN .EQ. 0.D0) GO TO 999
C
      DO 10 I=1,NS
      A(I) = A(I) / AN
   10 B(I) = B(I) / AN
C
      BN = B(NS)
C
C   FORM STATE VECTOR TIME DERIVATIVES AND PUT INTO YDOT
C   BEGINNING WITH LOCATION L IN YDOT.
C
C         CARRY OUT MATRIX OPERATIONS CALLED FOR BY EQUATION II-69
C
      DO 20 I=2,NS
C
         J = NS-I+1
```

```
          K = L+I-2                                                             0 8142
      IF (I .EQ. NS) GO TO 25                                                   C 8143
C                                                                               0 8144
   20 YJ(K) = -A(J)*Y(L) + Y(K+1) + (B(J)-A(J)*XN)*U                            0 8145
   25 YJ(K) = -A(J)*Y(L) + (B(J)-A(J)*XN)*U                                     C 8146
C                                                                               0 8147
      X = Y(L) + XN*U                                                           C 8148
C                                                                               C 8149
      RETURN                                                                    0 8150
C                                                                               0 8151
  999 CALL PAGEND                                                               0 8152
      WRITE (NIT,1001)                                                          0 8153
 1001 FORMAT (///,10X,33HCOEFFICIENT OF HIGHEST ,                               0 8154
     *              /,10X,32HDERIVATIVE OF DENOMINATOR CANNOT ,                 0 8155
     *             /,10X,17HBE EQUAL TO ZERO. ,                                 0 8156
     *             //,10X,16HPROGRAM STOPPED.)                                  0 8157
C                                                                               0 8158
      STOP                                                                      0 8159
      END                                                                       C 8160
```

```
      SUBROUTINE QCSORT (RR,RI,N,KR,KC,KZ,RERT,CMPR)                            0 8161
C                                                            DEBUG #  70        0 8162
      IMPLICIT REAL*8(A-H,O-Z)                                                  0 8163
C                                                                               0 8164
C   SUBROUTINE RECEIVES QR ROOT OUTPUT OF THE FORM,                             0 8165
C                     RR(I),RI(I),I=1,N                                         0 8166
C             AND PLACES REAL ROOTS (INCLUDING ZEROS) INTO                      C 8167
C             MATRIX RERT (SIZE KR). THEN PLACES THE                            0 8168
C             COMPLEX PAIRS (A(I) +J B(I)) INTO                                 C 8169
C             MATRIX CMPR-- COMPLEX PAIR ORDER IS                               C 8170
C                                                 CMPR(I) = A(I)                C 8171
C                                                 CMPR(I+1) = B(I)              0 8172
C             SAVES ONLY REAL AND POSITIVE IMAG PARTS IN CMPR.                  0 9:73
C             SIZE OF CMPR IS 2 * KC.                                           0 8174
C                                                                               0 8175
C             -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                        0 8176
C                                                                               0 8177
C   RR      = INPUT  ARRAY OF REAL PARTS.  SIZE IS N                            0 8178
C   RI      = INPUT  ARRAY OF IMAGINARY PARTS.  SIZE IS N.                      0 8179
C   N       = INPUT  SIZE OF RR AND RI.  NUMBER OF QR ROOTS.                    0 8180
C   KR      = OUTPUT SIZE OF RERT.  NUMBER OF REAL ROOTS,                       0 8181
C                                 (INCLUDING ZEROS).                            C 8182
C   KC      =                     NUMBER OF COMPLEX PAIRS.                      0 8183
C   KZ      = OUTPUT NUMBER OF ZEROS.                                           0 8184
C                                                                               0 8185
C   RERT    = OUTPUT REAL ROOT ARRAY.  SIZE KR.                                 0 8186
C   CMPR    = OUTPUT  COMPLEX PAIR ARRAY. SIZE 2*KC.                            0 8187
      DIMENSION RR(1),RI(1),RERT(1),CMPR(1)                                     0 8188
      KR = 0                                                                    C 8189
      KC = 0                                                                    0 8190
      KZ = 0                                                                    C 8191
C                                                                               0 8192
      DO 10 I=1,N                                                               0 8193
         IF (RI(I) .EQ. 0.D0) GO TO 5                                           0 8194
         IF (RI(I) .LT. 0.D0) GO TO 10                                          0 8195
```

```
            KC = KC+1
            L  = 2*KC
          CMPR(L-1) = RR(I)
          CMPR(L  ) = RI(I)
      GO TO 10
    5 CONTINUE
            KR = KR+1
          RLRT(KR) = RR(I)
          IF (RR(I) .EQ. 0.D0) KZ = KZ+1
   10 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE RMVZRO (RR,NR)
C                                                           DEBUG #  71
      IMPLICIT REAL*8(A-H,O-Z)
C
C  SUBROUTINE REMOVES REAL ZEROS FROM REAL ROOT ARRAY.
C
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C  RR      = INPUT/OUTPUT ARRAY CONTAINING ALL REAL ROOTS.
C  NR      = INPUT/OUTPUT NUMBER OF REAL ROOTS.
C
      DIMENSION RR(1)
C
      K=0
      DO 10 I=1,NR
      IF (RR(I) .EQ. 0.D0) GO TO 10
      K=K+1
      RR(K) = RR(I)
   10 CONTINUE
      NR = K
      RETURN
      END
```

```
      SUBROUTINE SIFT (A,N,TOL)
C                                                           DEBUG #  72
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(1)
C
C  SUBROUTINE SEARCHES ARRAY, A, FOR SMALL VALUES OF A AND SETS
C              THESE SMALL VALUES TO 0.0.
C
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C  A       = INPUT,OUTPUT VECTOR ARRAY TO BE SCANNED FOR SMALL VALUES.
C  N       = INPUT  SIZE OF A.
C  TOL     = INPUT  TOLERANCE.  IF (A(I) .LT. TOL  A(I) = 0.0
C
      DO 10 I=1,N
      IF (CABS(A(I)) .LT. TOL) A(I) = 0.D0
   10 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SFREQ2 (NNR,ICN,NDR,ICD,GAIN,
     1                    FDRN,FDNC,FDR ,FDDC ,
     2                    FMIN,FMAX,TITLE)
C                                                        DEBUG #  73
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SAVEG, SAVEP, SAVED, SAVEA
      REAL*4 FMIN , FMAX , TITLE
C
C SUBROUTINE DETERMINES S-PLANE FREQUENCY RESPONSE
C USING VARIABLE INCREMENTING TECHNIQUES.
C
C  FREQUENCY RESPONSE SAVED IN COMMON BLOCK /PSTUFF/
C
C                -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C   NNR     = INPUT NUMERATOR REAL ROOT COUNT.
C   ICN     = INPUT NUMERATOR COMPLEX PAIR ROOT COUNT.
C   NDR     = INPUT DENOMINATOR REAL ROOT COUNT.
C   ICD     = INPUT DENOMINATOR COMPLEX PAIR ROOT COUNT.
C   GAIN    = INPUT BODE GAIN.
C   FDRN    = INPUT NUMERATOR REAL ROOTS (INCLUDING ZEROS).
C   FDNC    = INPUT NUMERATOR COMPLEX PAIRS.
C   FDR     = INPUT DENOMINATOR REAL ROOTS (INCLUDING ZEROS).
C   FDDC    = INPUT DENOMINATOR COMPLEX PAIRS.
C   TITLE   = INPUT 30 CHARACTER ALPHANUMERIC TITLE.
C
      COMPLEX*16 PROD,FDEN,FNUM
      DIMENSION FDRN (1) , FDNC (1) , FDR (1)  , FDDC (1) ,
     1           TITLE(1) , KD(207) , TABG(20) , TABGP(31),
     2          TABZ(30) ,TABDN(31) , PROD(1)  ,
     3          FNUM(100), FDEN(100)
      COMMON /PSTUFF/
     *              SAVEG(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE
      COMMON /KOSIZE/
     1                KR, KNI, KNX, KV1, KV2, KVA
      COMMON /  LV1 /
     C                V1  (100), V2  (100), V3  (100)
      COMMON /DEBUG/ DEBUG(120)
      LOGICAL DEBUG,LDBG
      EQUIVALENCE (LDBG,DEBUG(73))
C
C
      DATA KD/207/
      DATA KSAVE/500/
      DATA NOT/ 0 /
C   TABC = CHOLE CONSTANT TABLE
```

```
C
      DATA KTABG,KTABUP,KTABDN/18,31,29/
C
      DATA TABUP /
     1      0.6000000, 0.7000000, 0.7500000, 0.8000000, 0.8400000,
     2      0.8800000, 0.9000000, 0.9200000, 0.9400000, 0.9600000,
     3      0.9650000, 0.9700000, 0.9750000, 0.9800000, 0.9840000,
     4      0.9880000, 0.9900000, 0.9920000, 0.9940000, 0.9960000,
     5      0.9980000, 0.9988000, 0.9995000, 0.9997500, 0.9999000,
     6      0.9999200, 0.9999400, 0.9999600, 0.9999800, 0.9999900,
     7      1.0000000/
C
      DATA TABG / 1.0000, 1.1000, 1.2500, 1.4000, 1.5000, 1.8000,
     1            2.0000, 2.2000, 2.5000, 2.8000, 3.2000, 3.8000,
     2            4.5000, 5.2000, 6.2000, 7.0000, 7.5000, 8.9000,
     3            0.0000, 0.0000/
C
C     TABUP   --- LEADING UP TO THE DAMPED NATURAL FREQUENCIES.
C
      WMAX  = 0.D0
      FOPHI = 0.D0
      WI    = 0.D0
C
      KWCT = 0
      KOPH = 0
      KPRINT = 1
      KSAVE = 0
C
      NRN = NNR
      NCN = ICN
      NRD = NDR
      NCD = ICD
      FK = GAIN
C
C
      DO 110 I=1,KTABDN
      J = KTABUP-I
  110 TABDN(I) = 1.D0/TABUP(J)
C
C               TABDN --- LEADING AWAY FROM DAMPED NATURAL FREQUENCIES
C               LNCTR --- LINE COUNTER
C
  120 LNCTR = 50
      JX   = 10
C
C              NULL WD
C
      DO 125 I= 1,KWD
  125 WD(I) = 0.D0
  130 CONTINUE
      DO 140 I=1,KR
      FNUM(I) = (0.D0,0.D0)
      FDEN(I) = (0.D0,0.D0)
  140 CONTINUE
C
C               FMIN = LOWER LIMIT.
C               SAVE IT AND DESTROY SAVED IF NEEDED
      SAVED = DBLE(FMIN)
C
C               COMPUTE  NUMERATOR DAMPED NATURAL FREQUENCIES.
```

```
C            DELETE FREQUENCIES WHICH ARE HIGHLY DAMPED
C              IF(2*ZETA**2.GT.1.0) SKIP ASSOCIATED FREQUENCY
C
      I = 0
      IF (NCN) 180, 180, 130
  130 NTCIN = NCN*2
      DO 170 J=1,NTCIN,2
      AHLE = FCNC(J) * FCNC(J+1)
      EAKER = FCNC(J+1) * DSQRT(1.DO - FCNC(J)**2)
      TEMP  = (EAKER)**2 - (AHLE)**2
      IF (TEMP) 170, 170, 160
  160 I      = I+1
      WD(I) = EAKER
  170 CONTINUE
C
C                  COMPUTE DENOMINATOR DAMPED NATURAL FREQUENCIES.
C
  180 IF (NCD) 240, 220, 190
  190 NTCD = NCD*2
C
      DO 210 J=1,NTCD,2
      AHLE   = FCDC(J) * FCDC(J+1)
      EAKER  = FCDC(J+1) * DSQRT(1.DO - FCDC(J)**2)
      TEMP   = (EAKER)**2 - (AHLE)**2
      IF (TEMP) 210, 210, 200
  200 I      = I+1
      WD(I) = EAKER
  210 CONTINUE
  220 KCOUNT = I
C
C                  THERE ARE KCOUNT FREQUENCIES.
C                  SORT THEM IN INCREASING MAGNITUDE.
C
      IF (KCOUNT - 1) 240, 350, 250
  240 J=1
      GO TO 370
  250 DO 270 J=1,KCOUNT
      DO 270 I=J,KCOUNT
      IF (WD(J) .LE. WD(I)) GO TO 270
      TEMP  = WD(J)
      WD(J) = WD(I)
      WD(I) = TEMP
  270 CONTINUE
C
C                  SORT COMPLETE.
C                  CHECK FOR EQUAL FREQUENCIES.
C                  IF SO, ELIMINATE ONE.
C
  280 I = 1
      J = 2
C
  290 IF (WD(I) - WD(J)) 300, 320, 340
  300 I = I+1
      J = J+1
  310 IF (KCOUNT - J) 350, 350, 290
  320 DO 330 K=J,KCOUNT
      WD(K-1) = WD(K)
  330 CONTINUE
      WD(KCOUNT) = 0.DO
      KCOUNT = KCOUNT - 1
      GO TO 310
  340 CALL PAGEHD
      WRITE (NOT,1313)
 1313 FORMAT (//10X,23HTHE SORT ROUTINE FAILED /
     1             10X,16HPROGRAM STOPPED. )
      STOP
  350 CONTINUE
      IF (LEQU) CALL WRITE(WD,I,KCOUNT,6H  WD  ,1)
C
C
C
  360 I = 1
      J = 1
      IF (WD(1) .GT. 0.D0) GO TO 430
  370 W = TABG(J) * SAVLO
      IF (W .GT. DBLE(FMAX)) GO TO 400
      IF (KTABG .GT. J) GO TO 410
      SAVLO = SAVLO * 10.D0
      J  = 1
      KK = 1
      GO TO 840
C
C              THE SHOW IS OVER.
C              GETOUT.
C
  400 KK = 6
      GO TO 1490
  410 J  = J+1
      KK = 1
      GO TO 840
C
C              ENTRY POINT FOR LOOPING ON FREQUENCY INCREMENTING.
C
  420 CONTINUE
      GO TO (370, 450, 500, 610, 660, 1490),KK
  430 IF (WD(I) .GT. SAVLO) GO TO 450
      I=I+1
      GO TO 430
  450 IF (TABG(J)*SAVLO - TABUP(1)*WD(I)) 460, 470, 490
  460 W = TABG(J)*SAVLO
      IF (KTABG .GT. J) GO TO 480
      SAVLO = SAVLO * 10.D0
      J  = 1
      KK = 2
      GO TO 830
  480 J  = J+1
      KK = 2
      GO TO 830
  490 J  = 1
  500 IF (J - KTABUP) 520, 510, 530
  510 W = TABUP(J) * WD(I)
      KPRINT = 2
      J  = J+1
      KK = 3
      GO TO 430
  520 W  = TABUP(J) * WD(I)
      J  = J+1
      KK = 3
      GO TO 830
  530 IF (WD(I+1) .GT. 0.D0) GO TO 550
```

```
C
C                 THE LAST FREQUENCY IS I.
C                 MAKE I+1 A DUMMY.
C
      WD(I+1) = FMAX * TABDN(KTABDN)
  550 IF (TABUP(JX)*WD(I+1) - WD(I)) 560, 640, 650
  560 J = JX
  570 IF (TABUP(J)*WD(I+1)  - WD(I)) 580, 590, 600
  580 J = J+1
      GO TO 570
  590 J = J+1
  600 I = I+1
  610 IF ((J-KTABUP) .EQ. 0) KPRINT = 2
      IF (J .GT. KTABUP) GO TO 650
  620 W = TABUP(J)*WD(I)
      J  = J+1
      KK = 4
      GO TO 830
  630 J = 1
      GO TO 530
  640 J = JX+1
      GO TO 600
  650 J = 1
  660 IF (J .GT. KTABDN) GO TO 690
      IF (TABDN(J)*WD(I) - TABUP(JX)*WD(I+1)) 680, 740, 750
  680 W = TABDN(J) * WD(I)
      J  = J+1
      KK = 5
      GO TO 830
  690 IF (TABUP(1)*WD(I+1) - TABDN(KTABDN)*WD(I)) 700, 700, 760
  700 J = 1
  710 IF (TABUP(J)*WD(I+1) - TABDN(KTABDN)*WD(I)) 720, 730, 730
  720 J = J+1
      GO TO 710
  730 I = I+1
      GO TO 610
  740 J = JX+1
      GO TO 600
  750 J = JX
      GO TO 500
  760 IF (TABDN(KTABDN)*WD(I) - TABS(KTABS)*SAVED) 770, 810, 820
  770 J = 1
  780 IF (TABDN(KTABDN)*WD(I) - TABS(J)*SAVED) 790, 790, 800
  790 I = I+1
      GO TO 450
  800 J = J+1
      GO TO 780
  810 SAVED = SAVED * 10.00
      J  = 1
      I  = I+1
      GO TO 450
  820 SAVED = SAVED * 10.00
      GO TO 760
  830 IF (W .GT. FMAX) GO TO 1490
      IF(LP30) WRITE(NOUT,1030)) W,WD(I),SAVED,I,J,KK,JX,NACT
 1030 FORMAT (' W =',E15.8,' WD(I) =',E15.8,' SAVED =',E15.8,' I =',I3,
     *          ' J =',I3,' KK =',I3,' JX =',I3,' NACT =',I3)
  840 J1 = 1
  850 IF (KK-4) 1210, 910, 860
C
```

```
C                 EVALUATE NUMERATOR REAL ROOTS
C
  860 DO 900 I1=1,NRN
      IF (FERN(I1) .EQ. 0.D0) GO TO 880
  870 FNUM(J1) = DCMPLX(1.D0,FERN(I1)*W)
      GO TO 890
  880 FNUM(J1) = DCMPLX(0.D0,W)
  890 J1 = J1+1
  900 CONTINUE
  910 IF (NCN) 1220, 940, 920
C
C                 EVALUATE NUMERATOR COMPLEX PAIRS
  920 DO 930 I1 = 1,NTOTN,2
C
C                           REAL PART
C                           IMAGINARY PART
C
      FNUM(J1) = DCMPLX(1.D0 - W**2 / (FBNC(I1+1))**2 ,
     1                     (2.D0* FBNC(I1)*W)/ FBNC(I1+1))
      J1 = J1+1
  930 CONTINUE
C
C                 REPEAT THE ABOVE PROCEDURE FOR DENOMINATOR
C
  940 J1 = 1
  950 IF (NRD) 1230, 1010, 960
  960 DO 1000 I1=1,NRD
      IF (FDR(I1).EQ. 0.D0) GO TO 980
      FDEN(J1) = DCMPLX(1.D0 , FDR (I1)*W)
      GO TO 990
  980 FDEN(J1) = DCMPLX(0.D0 ,W)
  990 J1 = J1+1
 1000 CONTINUE
 1010 IF (NCD) 1240, 1040, 1020
 1020 DO 1030 I1 = 1,NTOTD,2
      ALPHA = 1.D0 - W**2 / (FBDC(I1+1))**2
      BETA  = 2.D0 * FBDC (I1) * W / FBDC (I1+1)
      IF (ALPHA .LT. 1.D-20 .AND. BETA .EQ. 0.D0) BETA = 1.0D-10
      FDEN(J1) = DCMPLX(ALPHA,BETA)
      J1 = J1+1
 1030 CONTINUE
C
C                 EVALUATE  F(S) WITH COMPLEX ARITHMETIC ROUTINE.
C
 1040 KN        = NRN+NCN
      KD        = NRD+NCD
      PROD(1) = DCMPLX(1.D0,0.D0)
      IF (KN .LE. 0) GO TO 1090
      IF (KD .LE. 0) GO TO 1130
      IF (KN .GE.KD) GO TO 1110
C
C                 FACTORS IN DENOMINATOR EXCEED THOSE IN NUMERATOR.
C
      DO 1080 I1=1,KN
      PROD(1) = FNUM(I1) * PROD(1)/FDEN(I1)
 1080 CONTINUE
 1090 K = KN+1
      DO 1100 I1=K,KD
      PROD(1) = PROD(1)/FDEN(I1)
 1100 CONTINUE
```

```
      GO TO 1150                                                        0 8598
C                                                                       0 8599
C                FACTORS IN NUMERATOR EXCEED THOSE IN DENOMINATOR.      0 8600
C                                                                       0 8601
 1110 DO 1120 II=1,KD                                                   0 8602
      PROD(I) = FNUM(II)*PROD(I)/FDEN(II)                               0 8603
 1120 CONTINUE                                                          0 8604
      IF (KN .LE. KD) GO TO 1150                                        0 8605
 1130 K = KD+1                                                          0 8606
      DO 1140 II=K,KN                                                   0 8607
      PROD(I) = PROD(I)*FNUM(II)                                        0 8608
 1140 CONTINUE                                                          0 8609
 1150 PROD(I) = PROD(I) * DCMPLX(FK,0.D0))                              0 8610
C                                                                       0 8611
C               EVALUATION OF F(S) IS NOW COMPLETE.                     0 8612
C               NORMAL COMPUTED FORM -- F(JW) = ALPHA + BETA.           0 8613
C               ALPHA = REAL PART.                                      0 8614
C               BETA  = IMAGINARY PART.                                 0 8615
C                                                                       0 8616
C          -----CARTESIAN FORM (X,Y)-----                               0 8617
      ALPHA = DREAL (PROD(I))                                           0 8618
      BETA  = DIMAG (PROD(I))                                           0 8619
C                                                                       0 8620
C               IN POLAR FORM -- F(JW) = (AR,PHI).                      0 8621
C               AR  IS AMPLITUDE                                        0 8622
C               PHI IS PHASE ANGLE                                      0 8623
C                                                                       0 8624
C          -----POLAR FORM-----                                         0 8625
      RED =  (ALPHA**2 + BETA**2)                                       0 8626
      AR  =  DSQRT(RED)                                                 0 8627
      PHI = 0.D0                                                        0 8628
      IF (AR .NE. 0.D0) PHI = DATAN2(BETA,ALPHA)                        0 8629
C               CONVERT PHI FROM RADIANS TO DEGREES.                    0 8630
      DPHI = PHI * 57.29580D0                                           0 8631
      IF (DPHI .LT. 0.D0) DPHI = DPHI + 360.D0                          0 8632
C                                                                       0 8633
C         -----PRINT FREQUENCY RESPONSE DATA-----                       0 8634
C                                                                       0 8635
C                                                                       0 8636
C              SET OUTPUT PARAMETERS                                    0 8637
C              CONVERT AR TO LOG BASE 10 AND DECIBELS.                  0 8638
C                                                                       0 8639
 1260 IF (AR .NE. 0.D0) GO TO 1270                                      0 8640
      BELL = -20.D0                                                     0 8641
      GO TO 1280                                                        0 8642
 1270 BELL = DLOG(AR)                                                   0 8643
 1280 DBELL = BELL * 8.68588961D0                                       0 8644
      PHI = DPHI / 57.29580D0                                           0 8645
      W1 = W / 6.2831853D0                                              0 8646
C                                                                       0 8647
 1285 IF (LNCTR - 50) 1320, 1290, 1290                                  0 8648
 1290 CALL PAGEHD                                                       0 8649
      WRITE (NOT,158) (TITLE(I3),I3=1,20)                               0 8650
  158 FORMAT (/,10X20A4,/)                                              0 8651
      WRITE (NOT,159)                                                   0 8652
  159 FORMAT (1H0,20X,9HFREQ/RAD/SEC      FREQ/HERTZ         REAL      I 0 8653
     1MAG              AMP          DECIBELS   RAD        DEG /)        0 8654
      LNCTR = 1                                                         0 8655
      IF (KPRINT .EQ. 1) GO TO 1310                                     0 8656
      KPRINT = 1                                                        0 8657
      WRITE (NCT,1620) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI
 1620 FORMAT (17H           *******D16.6,D14.6,D15.6,D14.6,D16.6,F10.3,
     1    F9.4,F10.4,9H ********)
      GO TO 1330
 1310 WRITE (NCT,162) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI
  162 FORMAT (9X,D24.6,D14.6,D16.6,D14.6,D16.6,F10.3,F9.4,F10.4)
      GO TO 1370
 1320 IF (KPRINT .EQ. 1) GO TO 1350
      WRITE (NCT,1620) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI
      KPRINT = 1
 1330 KWCT = 1
      IF (WO(KWCT+1) .LE.0.D0) GO TO 1370
      RAT = WO(KWCT) / WO(KWCT+1)
      JX = 2
      IF (RAT .LE. 0.42D0) GO TO 1370
C           GET MIDPOINT LOCATION BETWEEN RESONANCE POINTS
      RAT = (1.+RAT)/2.D0
      DO 1340 KKK=3,30
      X = RAT - TABUP(KKK)
      IF(X.LE.0.0D0) GO TO 1370
      JX = KKK
 1340 CONTINUE
      GO TO 1360
 1350 WRITE (NCT,162) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI
 1360 ENCTR = ENCTR + 1
 1370 CONTINUE
C
C-----SAVE DATA TO PLOT
C
      KSAVE = KSAVE + 1
      SAVEC(KSAVE) = W
      SAVED(KSAVE) = DBELL
      SAVEF(KSAVE) = DPHI
      SAVEA(KSAVE) = AR
C
C
 1480 CONTINUE
C
C----------CONTINUE FREQUENCY SWEEP UNTIL LIMITS ARE EXHAUSTED.
C
      IF (W1 .LT. DBLE(FMAX) .AND. KSAVE .LT. KDSAVE) GO TO 420
C
C          NORMAL TERMINATION LOGIC.
C
 1490 CONTINUE
C
      KSAVE = KSAVE-1
C
      RETURN
C                ERROR EXITS
C
 1210 CALL PAGEHD
      WRITE (NCT,135) NRN
      GO TO 1490
C
 1220 CALL PAGEHD
      WRITE (NCT,137) NCN
      GO TO 1490
C
 1230 CALL PAGEHD
```

```
      WRITE (NCT,139) NRD
      GO TO 1450
C
C               ERROR EXIT FORMATS
C
 1240 CALL PAGEHD
      WRITE (NCT,141) NCD
  135 FORMAT (54HDATA FOR NRN IS INCORRECT IN SUBROUTINE SFREQ2, NRN = ,
     1        I5)
  137 FORMAT (54HDATA FOR NCN IS INCORRECT IN SUBROUTINE SFREQ2, NCN = ,
     1        I5)
  139 FORMAT (54HDATA FOR NRD IS INCORRECT IN SUBROUTINE SFREQ2, NRD = ,
     1        I5)
  141 FORMAT (54HDATA FOR NCD IS INCORRECT IN SUBROUTINE SFREQ2, NCD = ,
     1        I5)
C
C
      STOP
      END
```

```
      SUBROUTINE RLOCUS (PPR,QQR,SCL,SR,SI,NP,NQ,THETAO,
     1                   XMIN,XMAX,YMAX,ALOC)
C
C                                                        DEBUG #   74
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 K
      REAL*4 XSAVE, YSAVE, SAVEQ, SAVEP, SAVED, SAVEA
C
      COMPLEX*16 S,P,Q,SQ,ERR,DCMPLX
      INTEGER OUT
      REAL SNGL,SIGN
      LOGICAL START
C
C     SUBROUTINE DETERMINES ROOT LOCI FOR A SINGLE ROOT.
C
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C  PPR      = INPUT NUMERATOR POLYNOMINAL COEFFICIENTS.
C  QQR      = INPUT DENOMINATOR POLYNOMINAL COEFFICIENTS.
C             NOTE....  ALL POLY COEFFICIENTS ARE IN ASCENDING ORDER.
C  SCL      = INPUT SCALE FACTOR.  NORMALLY = 1.
C  SR       = INPUT REAL PART OF STARTING ROOT.
C  SI       = INPUT IMAG PART OF STARTING ROOT.
C  NP       = INPUT SIZE OF PPR.
C  NQ       = INPUT SIZE OF QQR.
C  XMIN     = INPUT MINIMUM ADMISSIBLE REAL ROOT VALUE.
C  XMAX     = INPUT MAXIMUM ADMISSIBLE REAL ROOT VALUE.
C  YMAX     = INPUT MAXIMUM ADMISSIBLE IMAG ROOT VALUE.
C  ALOC     = INPUT PHASE CONTROL PARAMETER
C                   ALOC = +1 ----+180 DEGREES PHASE (POSITIVE GAIN K)
C                   ALOC = -1 ----   0  DEGREES PHASE (NEGATIVE GAIN K)
C
      DIMENSION PPR(1),QQR(1)
      DIMENSION PAR(100), QAR(100)
      DIMENSION XSAVE(500),YSAVE(500)
```

```
C                                                                         0 8772
      COMMON /FSTUFF/                                                     0 8773
     *                 SAVEQ(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE 447 8774
      COMMON /LV1   /                                                     0 8775
     C                 V1  (100), V2  (100), V3  (100)                  426 8776
      EQUIVALENCE (V1(1),PAR(1)),(V2(1),QAR(1))                           0 8777
      EQUIVALENCE (SAVEQ(1),XSAVE(1)), (SAVEP(1),YSAVE(1))                0 8778
C                                                                         0 8779
      DATA OUT, KDSAVE /                                                  0 8780
     1       6 , 500   /                                                456 8781
C                                                                         0 8782
C                                                                         0 8783
C                                                                         0 8784
C                                                                         0 8785
      KSAVE = 1                                                           0 8786
C*******************************************************************      0 8787
C     ALOC SHOULD BE +1 FOR A 180 DEG LOCUS AND -1 FOR A 0 DEG LOCUS.     0 8788
C*******************************************************************      0 8789
      PI=3.141592653589793D0                                              0 8790
      ATR=180.D0/PI                                                       0 8791
C*******************************************************************      0 8792
C     FIND SCALE FACTOR (SCL) IF NOT SPECIFIED.                           0 8793
C     IF THE INPUTTED SCL IS POSITIVE, THAT VALUE WILL BE USED TO         0 8794
C     SCALE THE TWO POLYNOMIALS.  IF THE SCL IS NEGATIVE, A SCALE         0 8795
C     FACTOR WILL BE ESTIMATED FROM THE SIZE OF THE COEFFICIENTS OF       0 8796
C     THE POLYNOMIALS BY A LEAST SQUARES METHOD.                          0 8797
C*******************************************************************      0 8798
      IF(SCL.GT.0.D0)GO TO 100                                            0 8799
      CALL FIT(NP,PPR,SLOPEP)                                             0 8800
      CALL FIT(NQ,QQR,SLOPEQ)                                             0 8801
      SCL=1.D1**(-(NP*SLOPEP+NQ*SLOPEQ)/(NP+NQ))                          0 8802
C*******************************************************************      0 8803
C     WRITE HEADINGS                                                      0 8804
C*******************************************************************      0 8805
  100 CONTINUE                                                            0 8806
      NP1=NP-1                                                            0 8807
      NQ1=NQ-1                                                            0 8808
      CALL PAGEHD                                                         0 8809
      WRITE(OUT,38)                                                       0 8810
   38 FORMAT('1P(S) =')                                                   0 8811
      WRITE(OUT,40) PPR(1),(PPR(I+1),I,I=1,NP1)                           0 8812
      WRITE(OUT,39)                                                       0 8813
   39 FORMAT('0Q(S) =')                                                   0 8814
      WRITE(OUT,40) QQR(1),(QQR(I+1),I,I=1,NQ1)                           0 8815
   40 FORMAT (3X,D22.15,7X,3(1X,'+(',D22.15,'*S**',I2,')')/,              0 8816
     *                     4(1X,'+(',D22.15,'*S**',I2,')'))               0 8817
      WRITE(OUT,101) SCL                                                  0 8818
  101 FORMAT('0SCALE FACTOR =',D13.6)                                     0 8819
      WRITE(OUT,10)SR,SI                                                  0 8820
   10 FORMAT('0STARTING POINT = (',F18.11,') + I(',F18.11,')')            0 8821
      WRITE(OUT,15) XMIN,XMAX,YMAX,YMAX                                   0 8822
   15 FORMAT('0SCAN LIMITS:',D13.6,' < X < ',D13.6/' ',12X,'-',           0 8823
     *  D12.5,' < Y < ',D13.6)                                            0 8824
      IF(ALOC)16,16,17                                                    0 8825
   16 IDEG=0                                                              0 8826
      GO TO 18                                                            0 8827
   17 IDEG=180                                                            0 8828
   18 WRITE(OUT,19) IDEG                                                  0 8829
   19 FORMAT('0',43X,I3,' DEGREE LOCUS')                                  0 8830
      NLINE = 0                                                           0 8831
      CALL PAGEHD
      WRITE(IOUT,20)
   20 FORMAT(//13X,'GAIN',33X,'ROOTS',43X,'ERROR'//)
C***********************************************************************
C     SCALE EQUATIONS BY LETTING  S(NEW)=SCL*S(OLD)
C***********************************************************************
C
C     PAR WILL HOLD THE SCALED VERSION OF PPR
      ABP=DABS(PPR(NP))
      DO 120 I=1,NP1
      ABPP=PPR(I)/ABP
      NPI=NP-I
      DO 110 J=1,NPI
  110 ABPP=ABPP/SCL
  120 PAR(I)=ABPP*ALCC
      PAR(NP)=DSIGN(1.D0,PPR(NP))*ALCC
C
C     QAR WILL HOLD THE SCALED VERSION OF QQR
      ABQ=DABS(QQR(NQ))
      DO 140 I=1,NQ1
      ABQQ=QQR(I)/ABQ
      NQI=NQ-I
      DO 130 J=1,NQI
  130 ABQQ=ABQQ/SCL
  140 QAR(I)=ABQQ
      QAR(NQ)=DSIGN(1.D0,QQR(NQ))
C
      VK=ABQ/ABP
C     VK=RATIO OF THE MAGNITUDE OF THE LEADING COEFFICIENTS
      NQNP=NQ-NP
      DO 150 I=1,NQNP
  150 VK=VK*SCL
C     VK=SCALED RATIO
C***********************************************************************
C     INITIALIZE VALUES
C***********************************************************************
      NRTLOC=0
      START=.TRUE.
      SO=DCMPLX(SR/SCL,SI/SCL)
      RMINN=1.D-5
      RMIN=RMINN/SCL
      RMAX=1.D1/SCL
      RD=RMIN*1.01
      RO=RMIN
      CTM=1.D-6
      CTHETA=40.D0
C     THETA0 IS PASSED INTO THE SUBROUTINE FROM THE MAIN PROGRAM AND
C     HOLDS THE ANGLE (IN DEGREES) AT WHICH ALL SEARCHES ARE BEGUN.
      THETA1=THETA0
      THMAX=370.D0+THETA1
      THETA2=3600.
C     THETA2 IS USED IN THE ADJUST SEARCH RADIUS SECTION.  ITS INITIAL
C     VALUE IS ARBITRARY.
C***********************************************************************
C     START SEARCH
C***********************************************************************
  190 THETA=THETA1/ATR
      S=SO+DCMPLX(RO*DCOS(THETA),RO*DSIN(THETA))
C     S IS THE POINT TO BE EXAMINED
C     NOW EVALUATE P(S) AND Q(S)
```

```
      P=PAR(NP)
      Q=QAR(NQ)
      DO 200 I=1,NP1
  200 P=P*S+PAR(NP-I)
      DO 210 I=1,NQ1
  210 Q=Q*S+QAR(NQ-I)
C     FIND PHASE ANGLE OF P(S)/Q(S)
      AP=DREAL(P)
      BP=DIMAG(P)
      CQ=DREAL(Q)
      DQ=DIMAG(Q)
      THN=BP*CQ-AP*DQ
      THD=AP*CQ+BP*DQ
C     ARCTAN(THN/THD)=PHASE ANGLE OF P(S)/Q(S)
C     CHECK TO SEE IF THIS NEW PHASE ANGLE HAS CROSSED THE 180 DEG LINE.
C     FOR BOTH SPEED AND ACCURACY, THIS CHECK IS PERFORMED BY EXAMINING
C     THE SIGNS OF THN AND THD RATHER THAN BY USING THE ARCTAN FUNCTION.
      ISNNEW=SIGN(1.1,SNGL(THN))
      IF(START)GO TO 280
      IF(THD.GE.0.D0 .OR. ISNNEW.EQ.ISNOLD)GO TO 281
      DTHETA=-DTHETA
      GO TO 281
  280 START=.FALSE.
  281 ISNOLD=ISNNEW
      IF(DABS(DTHETA).LT.DTM) GO TO 310
      IF(THETA1.GT.THMAX)GO TO 300
      IF(DTHETA.LT.30.D0) DTHETA=DTHETA/2.D0
      THETA1=THETA1+DTHETA
      GO TO 190
C*******************************************************************
C     END SEARCH
C*******************************************************************
C     TWO RESULTS OF THE SEARCH ARE POSSIBLE.  IF NO ROOT LOCUS POINT
C     WAS FOUND, THE COMPUTER GOES TO STATEMENT 300.  IF A ROOT LOCUS
C     POINT WAS FOUND, THE COMPUTER GOES TO STATEMENT 310.
C*******************************************************************
C     NO ROOT WAS FOUND.
C*******************************************************************
  300 RD=RD/1.5D0
      ANGINC=109.47D0
      THETA1=THETA2
      S=SO
      IF(RD.GT.RMIN)GO TO 350
      IF(NPTLOC.EQ.0)GO TO 304
      WRITE(OUT,301) RMINN
  301 FORMAT('0THE LAST POINT PRINTED IS WITHIN ',F7.5,' OF A ROOT.')
      RETURN
  304 WRITE(OUT,305) RMINN
  305 FORMAT('0THE INITIAL POINT IS NOT WITHIN ',F7.5,' OF THE LOCUS.')
      RETURN
C*******************************************************************
C     A ROOT LOCUS POINT WAS FOUND.
C     EXAMINE SMOOTHNESS OF ROOT LOCUS CURVE. SEE IF THIS POINT WILL BE
C     USED.  IF THE CURVE BENDS TOO SHARPLY, POINT WILL NOT BE ACCEPTED.
C     THE PROGRAM WILL START AT THE OLD SO. SHRINK THE SEARCH RADIUS RD,
C     AND LOOK FOR A CLOSER POINT WHICH IS MORE IN LINE WITH THE CURVE.
C*******************************************************************
  310 CONTINUE
C     FIND ACUTE ANGLE BETWEEN THETA1 AND THETA2 (THDIF)
      THDIF1=DABS(THETA1-THETA2)
      THDIF=DMIN1(THDIF1,360.D0-THDIF1)
C     ADJUST SEARCH RADIUS IF THDIF IS LESS THAN 15 DEG OR
C     GREATER THAN 30 DEG
      IANG=THDIF/15.D0
      IF(IANG-1) 320,340,330
C
C     GO HERE IF THDIF .LT. 15. DEG
  320 RO=1.5D0*RO
      ANGINC=138.59D0
      IF(RO.LE.RMAX)GO TO 345
      RO=RMAX
C
C     GO HERE IF 15. .GE. THDIF . LT. 30. DEG
  340 ANGINC=120.D0
C
C     THIS STATEMENT IS REACHED ONLY IF THE ROOT LOCUS POINT IS ACCEPTED
  345 CONTINUE
      NRTLOC=NRTLOC+1
      CQCP=CDABS(Q)/CDABS(P)
      K=VK*CQCP*ALLC
      X=DREAL(S)*SCL
      Y=DIMAG(S)*SCL
      YY=DABS(Y)
C        K = OPEN LOOP GAIN
C        X = REAL COORDINATE OF ROOT ASSOCIATED WITH K
C        Y = IMAGINARY COORDINATE OF ROOT ASSOCIATED WITH K
      ERR=CHP*CQCP
      IF(DABS(THN/THD).LE.2.D-5)GO TO 271
      ANGL=ATR*DATAN2(THN,THD)
      WRITE(OUT,270) ANGL
  270 FORMAT(' MINIMUM VALUE OF DTHETA REACHED, PRESENT VALUE OF THETA I
     *S',F15.5)
  271 NLINE = NLINE + 1
      IF (NLINE .LT. 50) GO TO 314
      NLINE = 1
      CALL PAGEHD
      WRITE (OUT,20)
  314 WRITE(OUT,50)K,X,Y,ERR
      YSAVE(KSAVE) = Y
      XSAVE(KSAVE) = X
   50 FORMAT(5(6X,G18.9))
      IF ((X .LT. XMIN .OR. X .GT. XMAX .OR. YY .GT. YMAX)
     *        .OR. (KSAVE .GE. KOSAVE)) RETURN
      KSAVE = KSAVE + 1
      GO TO 350
C
C     GO HERE IF THDIF .GE. 30. DEG
  330 CONTINUE
      IF(NRTLOC.EQ.1)GO TO 340
      IF(RO.LE.RMIN)GO TO 340
      RO=RO/1.5D0
      ANGINC=109.47D0
      THETA1=THETA2
      S=SO
C****************************************************************
C     PREPARE TO LOOK FOR NEXT POINT ON ROOT LOCUS
C     THIS SECTION IS REACHED ONLY IF A ROOT LOCUS POINT WAS FOUND.
C****************************************************************
  350 THETA2=THETA1
      SO=S
      THETA1=THETA2-ANGINC                                                  0 9012
      THMAX =THETA2+ANGINC                                                  0 9013
      DTHETA=4.D1                                                           0 9014
      START=.TRUE.                                                          C 9015
      GO TO 190                                                             C 9016
      END                                                                   C 9017



      SUBROUTINE FIT(N,ARRAY,SLOPE)                                         C 9018
C                                                        DEBUG #  75        0 9019
      REAL*8 Y,SUMY,SUMIY,DLOG10,SLOPE,ARRAY(130)                           0 9020
C                                                                           C 9021
      M=0                                                                   C 9022
      ISUM=0                                                                0 9023
      ISQSUM=0                                                              C 9024
      SUMY=0.D0                                                             C 9025
      SUMIY=0.D0                                                            C 9026
      DO 4 I=1,N                                                            C 9027
      IF(ARRAY(I))1,4,2                                                     C 9028
    1 Y=DLOG10(-ARRAY(I))                                                   0 9029
      GO TO 3                                                               C 9030
    2 Y=DLOG10(ARRAY(I))                                                    C 9031
    3 M=M+1                                                                 C 9032
      ISUM=ISUM+I                                                           0 9033
      ISQSUM=ISQSUM+I*I                                                     0 9034
      SUMY=SUMY+Y                                                           C 9035
      SUMIY=SUMIY+I*Y                                                       0 9036
    4 CONTINUE                                                              C 9037
      SLOPE=(M*SUMIY-ISUM*SUMY)/(M*ISQSUM-ISUM*ISUM)                        C 9038
      RETURN                                                                0 9039
      END                                                                   0 9040



      FUNCTION DIMAG (Y)                                                    0 9041
C                                                        DEBUG #  76        0 9042
      COMPLEX*16 Y                                                          0 9043
      COMPLEX SY                                                            0 9044
      SY = Y                                                                C 9045
      DIMAG = DBLE (AIMAG(SY))                                              0 9046
      RETURN                                                                C 9047
      END                                                                   C 9048



      FUNCTION DREAL (Y)                                                    0 9049
C                                                        DEBUG #  77        0 9050
      COMPLEX*16 Y                                                          0 9051
      COMPLEX SY                                                            0 9052
      SY = Y                                                                0 9053
      GREAL = DBLE (REAL(SY))                                            0 9054
      RETURN                                                             0 9055
      END                                                                0 9056




      SUBROUTINE CANCEL (R)                                              0 9057
C                                                      DEBUG #   78      0 9058
      IMPLICIT REAL*8(A-H,O-Z)                                           0 9059
C THIS ROUTINE CANCELS OUT THE ZERO, REAL, AND THE COMPLEX ROOTS THAT AR 0 9060
C COMMON TO THE NUMERATOR AND DENOMINATOR OF THE TRANSFER FUNCTION R.    0 9061
C                                                                        0 9062
C --- R(1) = NUMBER OF REAL ROOTS IN THE NUMERATOR                       0 9063
C --- R(2) = NUMBER OF COMPLEX PAIRS IN THE NUMERATOR                    0 9064
C --- R(3) = NUMBER OF ZERO ROOTS IN THE NUMERATOR                       0 9065
C --- R(4) = NUMBER OF REAL ROOTS IN THE DENOMINATOR                     0 9066
C --- R(5) = NUMBER OF COMPLEX PAIRS IN THE DENOMINATOR                  0 9067
C --- R(6) = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                     0 9068
C --- R(7) = GAIN                                                        0 9069
C --- R(8)...R(I) = NUMERATOR REAL ROOTS ARRAY                           0 9070
C --- R(I+1)...R(J) = NUMERATOR COMPLEX PAIRS ARRAY                      0 9071
C --- R(J+1)...R(K) = DENOMINATOR REAL ROOTS ARRAY                       0 9072
C --- R(K+1...R(L) = DENOMINATOR COMPLEX PAIRS ARRAY                     0 9073
C                                                                        0 9074
      DIMENSION R(1)                                                     0 9075
      NR=R(1)+.000100                                                    0 9076
      NCP=R(2)+.000100                                                   0 9077
      MR=R(4)+.000100                                                    0 9078
      MCP=R(5)+.000100                                                   0 9079
      KK=7+NR+MR+2*(NCP+MCP)                                             0 9080
      N = 7 + NR                                                         0 9081
      IF((NR.EQ.0).OR.(MR.EQ.0)) GO TO 160                               0 9082
      J=8+NR+2*NCP                                                       0 9083
      K=J-1+MR                                                           0 9084
      NN=0                                                               0 9085
      DO 140 I=8,N                                                       0 9086
      II=I-NN                                                            0 9087
      DO 100 JJ=J,K                                                      0 9088
      JJJ=JJ                                                             0 9089
      IF(DABS(R(II)/R(JJJ)-1.D0).LT.1.D-7) GO TO 110                     0 9090
  100 CONTINUE                                                           0 9091
      GO TO 140                                                          0 9092
  110 DO 130 L=II,KK                                                     0 9093
      IF(L.GE.(JJJ-1)) GO TO 120                                         0 9094
      R(L)=R(L+1)                                                        0 9095
      GO TO 130                                                          0 9096
  120 R(L)=R(L+2)                                                        0 9097
  130 CONTINUE                                                           0 9098
      KK=KK-2                                                            0 9099
      J=J-1                                                              0 9100
      K=K-2                                                              0 9101
      NN=NN+1                                                            0 9102
      IF((NN.EQ.NR).OR.(NN.EQ.MR)) GO TO 150                             0 9103
  140 CONTINUE                                                           0 9104
  150 R(1)=R(1)-DFLOAT(NN)                                               0 9105
      R(4)=R(4)-DFLOAT(NN)                                               0 9106
      NR=NR-NN                                                           0 9107
      NR=NR-NN
  160 IF((NCP.EQ.0).OR.(NCP.EQ.0)) GO TO 230
      NNN=1+NR
      N=NNN-1+2*NCP
      J=N+1+NR
      K=J-1+2*NCP
      NN=0
      DO 210 I=NNN,N,2
      II=I-2*NN
      DO 170 JJ=J,K,2
      JJJ=JJ
      IF((R(II).EQ.R(JJJ)).AND.(R(II+1).EQ.R(JJJ+1))) GO TO 180
  170 CONTINUE
      GO TO 210
  180 DO 200 L=II,KK
      IF(L.GE.(JJJ-2)) GO TO 190
      R(L)=R(L+2)
      GO TO 200
  190 R(L)=R(L+4)
  200 CONTINUE
      KK=KK-4
      J=J-2
      K=K-4
      NN=NN+1
      IF((NN.EQ.NCP).OR.(NN.EQ.MCP)) GO TO 220
  210 CONTINUE
  220 R(2)=R(2)-DFLOAT(NN)
      R(5)=R(5)-DFLOAT(NN)
  230 X=DMIN1(R(3),R(6))
      R(3)=R(3)-X
      R(6)=R(6)-X
      RETURN
      END
```

```
      SUBROUTINE FORMB (KR, KC, RERT, CMPR, FBR, FBC)
C                                                         DCEUG #  79
      IMPLICIT REAL*8(A-H,O-Z)
CFORMB        FACTORED  FORM  TIME CONSTANTS, DAMPING AND FREQUENCY
C   CALLING SEQUENCE FOR SUBROUTINE FORMB IS AS FOLLOWS
C   CALL FORMB (KR, KC, RERT, CMPR, FBR, FBC)
C    KR -- COUNT OF REAL ROOTS, MAY OR MAY NOT INCLUDE ZEROS.
C     KC -- COUNT OF THE COMPLEX PAIRS OF ROOTS.
C      RERT -- STORAGE BLOCK CONTAING ALL REAL ROOTS.
C      CMPR -- STORAGE BLOCK CONTAINING COMPLEX PAIRS OF ROOTS
C       FBC -- FORM B COMPLEX PAIR BLOCK (OUTPUT FROM ROUTINE)
C        FBR -- FORM B REAL ROOT BLOCK (OUTPUT FROM ROUTINE)
C
      DIMENSION RERT(1),CMPR(1),FBR(1),FBC(1)
C
      IF (KR) 140, 140, 100
  100 DO 120 L = 1,KR
      IF (RERT(L)) 120, 110, 120
  110 FBR(L) = 0.0
      GO TO 130
  120 FBR(L) = -1.D0/ RERT(L)
  130 CONTINUE
  140 IF (KC) 170, 170, 150
  150 KK = 2*KC
      DO 160 L = 2,KK,2
      FBC(L) = DSQRT(CMPR(L-1)**2 + CMPR(L)**2)
  160 FBC(L-1) = -CMPR(L-1)/FBC(L)
  170 RETURN
      END
```

```
      SUBROUTINE DFORMB(KR,KC,RERT,CMPR,FBR,FBC,SCMPR,SF,ACCD,GB)
C
C                                                     DEBUG #  80
      IMPLICIT REAL*8(A-H,O-Z)
CDFORMB      FACTORED  FORM  TIME CONSTANTS, DAMPING AND FREQUENCY
      DIMENSION RERT(1),CMPR(1),FBR(1),FBC(1),SCMPR(1)
      DIMENSION GB(2)
      LOGICAL DEBUG,LEQU
      COMMON /CDEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG( 80))
C         KR = NUMBER OF REAL AND ZERO ROOTS IN RERT
C         KC = NUMBER OF COMPLEX PAIRS IN CMPR
C         KK = 2*KC
C         RERT = REAL AND ZERO ROOT ARRAY
C         CMPR = COMPLEX PAIR ARRAY
C         SF = 1.0
C         ACCD = GAIN FOR NUMERATOR
C         ACCD = 1.0 FOR DENOMINATOR
      GB(1)=ACCD
      LZ=0
      SCI=1.D0/SF
      IF(KR)140, 140, 100
  100 DO 130 L=1,KR
      IF(RERT(L))120, 110, 120
  110 FBR(L)=0.D0
C         COUNT ZERO ROOTS
      LZ=LZ+1
      GO TO 130
C         FBR = TIME CONSTANTS ASSOCIATED WITH REAL ROOTS
  120 FBR(L)=-1.D0/(RERT(L)*SCI)
C         COMPUTE PRODUCT OF REAL ROOTS
      GB(1)=-GB(1)*RERT(L)
  130 CONTINUE
  140 IF(KC)170, 170, 150
C         CYCLE THROUGH ALL COMPLEX PAIRS
  150 KK=2*KC
      DO 160 L=2,KK,2
C         CONTINUE GETTING PRODUCT OF ALL ROOTS
      GB(1)=GB(1)*(CMPR(L-1)**2+CMPR(L)**2)
      SCMPR(L-1)=CMPR(L-1)*SCI
C         FBC(L) = FREQUENCY ASSOCIATED WITH COMPLEX PAIR
C         FBC(L-1) = DAMPING ZETA ASSOCIATED WITH COMPLEX PAIR
C         FORM:
C              S**2 + 2*ZETA*OMEGA*S + OMEGA**2 = 0
C         FBC(L) = OMEGA
C         FBC(L-1) = ZETA
```

```
      FBC(L)=DSQRT(SCMPR(L-1)**2+(CMPR(L)*SCI)**2)                      0 9216
  160 FBC(L-1)= -SCMPR(L-1)/FBC(L)                                      0 9217
C          RETURN WITH PRODUCT OF ALL NON-ZERO ROOTS IN GB(1) AND GB(2) 0 9218
  170 GB(2)=GB(1)                                                       0 9219
      IF (LZ .EQ. 0) GO TO 180                                          C 9220
      GB(2)=GB(2)*SF**LZ                                                0 9221
  180 CONTINUE                                                          0 9222
      IF(.NOT.LEQU) RETURN                                              0 9223
      CALL WRITE(RLRT,1,KR,6H RLRT ,1)                                  C 9224
      CALL WRITE(CMPR,1,KK,6H CMPR ,1)                                  0 9225
      CALL WRITE(FBR ,1,KR,6H  FBR ,1)                                  0 9226
      CALL WRITE(FBC ,1,KK,6H  FBC ,1)                                  0 9227
      CALL WRITE(ACCD,1, 1,6H ACCD ,1)                                  0 9228
      CALL WRITE(  GB,1, 2,6H GB   ,1)                                  0 9229
      RETURN                                                            0 9230
      END                                                               0 9231




      SUBROUTINE RTOP (RTS,POLY,TEMP,KPLY)                              C 9232
C                                                          DEBUG #  81  0 9233
      IMPLICIT REAL*8(A-H,O-Z)                                          0 9234
CRTOP         TRANSFER FUNCTION ROOTS TO TRANSFER FUNCTION POLYNOMIALS  0 9235
C                                                                       0 9236
C           SUBROUTINE RTOP,  ROOTS CONVERTED TO POLYNOMIAL             C 9237
C                                                                       0 9238
C --- RTS(1) = NUMBER OF REAL ROOTS IN THE NUMERATOR                    C 9239
C --- RTS(2) = NUMBER OF COMPLEX PAIRS IN THE NUMERATOR                 0 9240
C --- RTS(3) = NUMBER OF ZERO ROOTS IN THE NUMERATOR                    C 9241
C --- RTS(4) = NUMBER OF REAL ROOTS IN THE DENOMINATOR                  C 9242
C --- RTS(5) = NUMBER OF COMPLEX PAIRS IN THE DENOMINATOR               0 9243
C --- RTS(6) = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                  0 9244
C --- RTS(7) = GAIN FACTOR                                              C 9245
C --- RTS(8)...RTS(I) = NUMERATOR REAL ROOTS ARRAY                      0 9246
C --- RTS(I+1)...RTS(J) = NUMERATOR COMPLEX ROOTS ARRAY                 C 9247
C --- RTS(J+1)...RTS(K) = DENOMINATOR REAL ROOTS ARRAY                  0 9248
C --- RTS(K+1)...RTS(L) = DENOMINATOR COMPLEX ROOTS ARRAY               0 9249
C --- POLY(1) = DEGREE OF THE NUMERATOR                                 C 9250
C --- POLY(2) = DEGREE OF THE DENOMINATOR                               0 9251
C --- POLY(3)...POLY(I) = ALL COEFFICIENTS OF NUMERATOR FOR ASCENDING PO 0 9252
C --- POLY(I+1)...POLY(J) = ALL COEFFICIENTS OF DENOMINATOR FOR ASCENDIN C 9253
C --- OF S                                                              0 9254
C --- TEMP = TEMPORARY STORAGE                                          0 9255
C --- NCD = NUMBER OF COMPLEX PAIRS IN DENOMINATOR                      C 9256
C --- NCN = NUMBER OF COMPLEX PAIRS IN NUMERATOR                        0 9257
C --- NDEN = TOTAL NUMBER OF DENOMINATOR ROOTS IN RTS ARRAY             0 9258
C --- NNUM = TOTAL NUMBER OF NUMERATOR ROOTS IN RTS ARRAY               0 9259
C --- NRD = NUMBER OF REAL ROOTS IN THE DENOMINATOR                     0 9260
C --- NRN = NUMBER OF REAL ROOTS IN THE NUMERATOR                       0 9261
C --- NZD = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                     0 9262
C --- NZN = NUMBER OF ZERO ROOTS IN THE NUMERATOR                       0 9263
C --- KPLY = DIMENSION SIZE OF POLY IN CALLING PROGRAM.                 0 9264
C                                                                       0 9265
      DIMENSION RTS(1),POLY(1),TEMP(1)                                  0 9266
      KTAPE = 6                                                         0 9267
      NRN=RTS(1) + 0.1D0                                                0 9268
      NCN=RTS(2) + 0.1D0                                                0 9269
      NZN=RTS(2) + 0.1D0
      NRD=RTS(4) + 0.1D0
      NCD=RTS(5) + 0.1D0
      NZD=RTS(6) + 0.1D0
      DO 100 I = 1,KPLY
      POLY(I)=0.D0
  100 TEMP(I) = 0.D0
C
      NNUM = 2 * NCN + NRN + NZN
      NDEN = 2 * NCD + NRD + NZD
      POLY(1)=NNUM
      POLY(2)=NDEN
C
      KP = 0
      POLY (3) = 1.D0
      IF (NRN) 510, 110, 130
  110 IF (NCN) 510, 120, 190
C             NUMERATOR IS GAIN TERM ONLY
  120 KP = NZN + 3
      POLY(3) = 0.D0
      POLY(KP) = RTS(7)
      KP = KP+1
      GO TO 250
C             NUMERATOR REAL ROOTS
  130 TEMP (2) = RTS(8)
      POLY (4) = TEMP (2)
      IF (NRN-1) 180, 180, 140
  140 DO 170 K = 2, NRN
      DO 150 K1 = 1, K
  150 TEMP (K1+1) = RTS(K+7) * POLY (K1+2) + POLY (K1+3)
      DO 160 K2 = 1, K
  160 POLY (K2+3) = TEMP (K2+1)
  170 CONTINUE
C
  180 IF (NCN) 510, 250, 190
C
C             INCLUDE THE NUMERATOR COMPLEX ROOTS
  190 KNR = NRN
      KC = NRN +8
      KCM = 2 * NCN + KC - 1
      DO 240 L = KC, KCM, 2
      TEM1 = 2.D0*RTS(L)/RTS(L+1)
      TEM2 = 1.D0/RTS(L+1)**2
      LL = L-6
      DO 220 L2 = 1, LL
      TEM3 = 0.D0
      IF (L2-1) 210, 210, 200
  200 TEM3 = TEM2 * POLY (L2+1)
  210 TEMP(L2+1) = POLY (L2+3) + TEM1 * POLY (L2+2) + TEM3
  220 CONTINUE
      KNR = KNR + 2
      DO 230 L3 = 1, KNR
  230 POLY (L3+3) = TEMP (L3+1)
  240 CONTINUE
C
C             ENTER GAIN FACTOR, ZERO ROOTS, RESTORE COEFFICIENTS
  250 KP = NZN + 4
      KS = 2 * NCN + NRN
      DO 260 J = 1, KS, 1
      POLY (KP) = TEMP (J+1) * RTS (7)
  260 KP = KP+1
      POLY (NZN+3) = 1.D0* RTS (7)
      IF (NZN) 290, 290, 270
  270 DO 280 J = 1, NZN, 1
  280 POLY (J+2) = 0.D0
C
C          PROCESS DENOMINATOR ROOTS, KP IS LOCATION FOR STORING FIRST
C          DENOMINATOR COEFFICIENT
C
  290 POLY (KP) = 1.D0
      IF (NRD) 310, 300, 340
  300 IF (NCD) 510, 310, 400
C          DENOMINATOR REAL ROOTS (KR IS LOCATION FOR FIRST ROOT)
  310 IF (NZD) 510, 390, 320
  320 KRIP = KP + NZD
      DO 330 I5 = KP,KRIP
      POLY(I5) =0.D0
  330 CONTINUE
      POLY(KRIP) = 1.D0
      GO TO 390
  340 KR = 2*NCN + NRN + 6
      POLY(KP+1) = RTS(KR)
      TEMP(2) = RTS(KR)
      IF (NRD-1) 390, 390, 350
  350 DO 380 K=2,NRD
      NC1 = KR+K-1
      DO 360 K1 = 1,K
      NC2 = KP+K1-1
  360 TEMP(K1+1) = RTS(NC1)*POLY(NC2) + POLY(NC2+1)
      DO 370 K2 = 1,K
      NC3 = KP+K2
  370 POLY(NC3) = TEMP(K2+1)
  380 CONTINUE
C
  390 IF (NCD) 510, 460, 400
C
C          PROCESS DENOMINATOR COMPLEX ROOTS
  400 KOR = NRD
      KC = 2*NCN + NRN + NRD +6
      KCM = 2* NCD + KC-1
      DO 450 L = KC, KCM, 2
      TEM1 = 2.D0*RTS(L)/RTS(L+1)
      TEM2 = 1.D0/RTS(L+1)**2
      LL= L-(2*NCN+NRN+6)
      DO 420 L2 = 1, LL
      NC5 = KP+L2-1
      TEM3 = 0.D0
      IF (L2-1) 420, 420, 410
  410 TEM3 = TEM2 * POLY (NC5-1)
  420 TEMP (L2+1) = POLY (NC5+1) + TEM1 * POLY (NC5) + TEM3
  430 CONTINUE
      KOR = KOR + 2
      DO 440 L3 = 1, KOR
      NC6 = KP+L3
  440 POLY (NC6) = TEMP (L3+1)
  450 CONTINUE
C
  460 KD = KP + NZD + 1
      KS = 2 * NCD + NRD
      DO 470 M = 1, KS, 1
      POLY (KD) = TEMP (K+1)
  470 KD = KD+1
      KD = KP+NZD-1
      POLY (KD+1) = POLY (KP)
      IF (NZD) 500, 500, 480
  480 DO 490 J = KP, KD, 1
  490 POLY (J) = 0.D0
C
  500 RETURN
C          ERROR COMMENT AND RETURN TO MAIN PROGRAM
  510 WRITE (KTAPE,1002)
 1002 FORMAT (68H1 A NEGATIVE COUNT OF ROOTS WAS ENCOUNTERED I RTOP.  PO
     1LYNOMIAL COULD NOT BE OBTAINED.)
      RETURN
      END
```

```
      SUBROUTINE TRFB  (ND,RX,KR,KC,KZ,FBR,FBC,GG,ZGV,KSIZE)
C
C                                                       DEBUG #  82
      IMPLICIT REAL*8(A-H,O-Z)
CTRFB          TRANSFER INPUT ROOTS FORMB AND FORMC
C    ND -- IF INPUT (0) WE HAVE NUMERATOR, IF INPUT (1) A DENOMINATOR
C      RX -- ENTIRE BLOCK (COUNTS,GAIN,ROOTS)
C       KR -- RUNNING COUNT OF ACCUMULATED REALS FOR ANY GIVEN CASE
C        KC -- SAME AS ABOVE BUT FOR COMPLEX
C         KZ -- COUNT OF ACCUMULATED ZEROS FOR ANY GIVEN CASE
C          FBR -- FORM (B) REAL STORAGE BLOCK
C           FBC -- FORM (C) COMPLEX STORAGE BLOCK
C             GG -- RUNNING GAIN TERM
C               ZGV -- IF OUTPUT OTHER THAN ZERO, ABSF(ZETA) EXCEEDED (1)
C                 KSIZE =  DIMENSIONED SIZE OF FBR AND FBC .
C  IT IS ASSUMED THAT THE COUNT FOR THE ACCUMULATED ROOTS WILL BE ZEROED
C  OUT AT THE BEGINNING OF EACH CASE.
C  ANOTHER TASK IN THE (MAIN) PROGRAM IS CHECKING THE (ZETA) FLAG.
      DIMENSION RX(1),FBR(1),FBC(1)
      IF (GG .EQ. 0.D0) RETURN
      IF (ND.GT.0) GO TO 90
      IF (RX(7) .EQ. 0.D0) GO TO 200
      GG=GG*RX(7)
   90 KRX1 = RX(1)+ 0.1D0
      KRX2 = RX(2)+ 0.1D0
      KRX3 = RX(3)+ 0.1D0
      KRX4 = RX(4)+ 0.1D0
      KRX5 = RX(5)+ 0.1D0
      KRX6 = RX(6)+ 0.1D0
      IF (ND) 100, 100, 110
  100 J = 7
      JCR = KRX1
      JCC = 2*KRX2
      JCZ = KRX3
      GO TO 120
  110 J = 7 + KRX1 + 2 * KRX2
      JCR = KRX4
      JCC = 2*KRX5
      JCZ = KRX6
  120 IF (JCR) 150, 150, 130                                                  0 9444
  130 DO 140 M = 1,JCR                                                         0 9445
      KR = KR+1                                                                0 9446
      L = J+M                                                                  0 9447
      FBR(KR) = RX(L)                                                          0 9448
  140 CONTINUE                                                                 0 9449
  150 IF (JCC) 190,190,160                                                     0 9450
  160 DO 180 M = 2,JCC,2                                                       0 9451
      KC = KC+1                                                                0 9452
      KK = 2*KC                                                                0 9453
      L = J+JCR+M                                                              0 9454
      FBC(KK-1) = RX(L-1)                                                      0 9455
      FBC(KK) = RX(L)                                                          0 9456
      IF (DABS(RX(L-1))- 1.D0) 180, 180, 170                                   0 9457
  170 ZOV = 1.D0                                                               0 9458
  180 CONTINUE                                                                 0 9459
  190 KZ = KZ+JCZ                                                              0 9460
      RETURN                                                                   0 9461
  200 GG=0.D0                                                                  0 9462
      KR=0                                                                     0 9463
      KC=0                                                                     0 9464
      KZ=0                                                                     0 9465
      DO 210 I=1,KSIZE                                                         0 9466
      FBR(I)=0.D0                                                              0 9467
  210 FBC(I)=0.D0                                                              0 9468
      RETURN                                                                   0 9469
      END                                                                      0 9470




      SUBROUTINE TIFF (NR,NC,NZ,KR,KC,KZ,G,RN,CN,RD,CD,R,KSIZE)                0 9471
C                                                                              0 9472
C                                                            DEBUG #  83       0 9473
      IMPLICIT REAL*8(A-H,O-Z)                                                 0 9474
C                                                                              0 9475
C --- NR = NUMBER OF REAL ROOTS IN THE NUMERATOR                               0 9476
C --- NC = NUMBER OF COMPLEX PAIRS IN THE NUMERATOR                            0 9477
C --- NZ = NUMBER OF ZERO ROOTS IN THE NUMERATOR                               0 9478
C --- KR = NUMBER OF REAL ROOTS IN THE DENOMINATOR                             0 9479
C --- KC = NUMBER OF COMPLEX PAIRS IN THE DENOMINATOR                          0 9480
C --- KZ = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                             0 9481
C --- G = GAIN                                                                 0 9482
C --- RN = NUMERATOR REAL ROOT ARRAY                                           0 9483
C --- CN = NUMERATOR COMPLEX PAIRS ARRAY                                       0 9484
C --- RD = DENOMINATOR REAL ROOT ARRAY                                         0 9485
C --- CD = DENOMINATOR COMPLEX PAIRS ARRAY                                     0 9486
C --- R = ARRAY CONTAINING NUMBER OF ROOTS AND ROOT ARRAYS.                    0 9487
C -KSIZE- = DIMENSIONED SIZE OF R IN CALLING PROGRAM.                          0 9488
C                                                                              0 9489
      DIMENSION RN(1),CN(1),RD(1),CD(1),R(1)                                   0 9490
      LOGICAL DEBUG,LEQU                                                       0 9491
      DATA NOT/6/                                                              0 9492
      COMMON /LDEBUG/ DEBUG(120)                                               0 9493
      EQUIVALENCE (LEQU,DEBUG( 83))                                            0 9494
      IF (G.EQ.0.D0) GO TO 80                                                  0 9495
      R(1)=NR                                                                  0 9496
      R(2)=NC                                                                  0 9497
      R(3)=N2                                                           0 9498
      R(4)=KR                                                           0 9499
      R(5)=KC                                                           C 9500
      R(6)=K2                                                           C 9501
      R(7)=G                                                            0 9502
      L=7+NR                                                            C 9503
      IF (NR.LE.0.00) GO TO 20                                          0 9504
      DO 10 I=8,L                                                       0 9505
   10 R(I)=RN(I-7)                                                      C 9506
   20 N=L+1                                                             C 9507
      L=L+2*NC                                                          0 9508
      IF(NC.LE.0) GO TO 40                                              C 9509
      DO 30 I=N,L                                                       0 9510
      J=I-N+1                                                           C 9511
   30 R(I)=CN(J)                                                        C 9512
   40 N=L+1                                                             C 9513
      L=L+KR                                                            0 9514
      IF(KR.LE.0) GO TO 60                                              0 9515
      DO 50 I=N,L                                                       C 9516
      J=I-N+1                                                           C 9517
   50 R(I)=RK(J)                                                        0 9518
   60 IF(KC.LE.0) GO TO 100                                             C 9519
      N=L+1                                                             C 9520
      L=L+2*KC                                                          0 9521
      DO 70 I=N,L                                                       0 9522
      J=I-N+1                                                           C 9523
   70 R(I)=CK(J)                                                        C 9524
      GO TO 100                                                         0 9525
   80 DO 90 I=1,KSIZE                                                   0 9526
   90 R(I)=0.0D0                                                        0 9527
  100 CONTINUE                                                          C 9528
      IF(.NOT.DEBUG) RETURN                                             0 9529
      WRITE(NOT,1000)                                                   0 9530
 1000 FORMAT (' SUBROUTINE TIFF ROOTS ARRAY IS ')                       0 9531
      CALL WRITES(R,1,KSIZE,1)                                          C 9532
      RETURN                                                            C 9533
      END                                                               0 9534



      SUBROUTINE GRDRVR (A,N,RR,RI,KR)                                  0 9535
C                                                   DEBUG #  84         0 9536
      IMPLICIT REAL*8(A-H,O-Z)                                          0 9537
C                                                                       0 9538
C    ----LATEST MOD ON 09/04/75                                         0 9539
C    ----WHEN ENTIRE PACKAGE WAS MADE INTO G-PRECISION                  0 9540
C                                                                       0 9541
CC   FOR EXTENDED PRECISION, G, VI MUST BE DECLARED REAL*16.            0 9542
C       ALL OTHERS DECLARED REAL*8.                                     0 9543
C                                                                       C 9544
C                                                                       0 9545
      DIMENSION A(KR,1), RR(1), RI(1)                                   0 9546
      DIMENSION G(100,100), VI(100)                                   410 9547
C                                                                       0 9548
      COMMON /ZWORK1/                                                   0 9549
     F                 W1(100,100), W2(100,100)                       432 9550
C                                                                       0 9551
```

```
      EQUIVALENCE (Q(1),W1(1))
      LOGICAL DEBUG,LEQU
      COMMON /LDEBUG/ DEBUG(120)
      EQUIVALENCE (LEQU,DEBUG(64))
      DATA NOT/6/
C
CCC  NOTE.. THE EQUIVALENCE STATEMENT MUST READ AS FOLLOWS
CCC           ((WHEN EXTENDED, OR Q-PRECISION))
CC    EQUIVALENCE (Q(1),W1(1)), (Q(5001),W2(1))
C
      IF(LEQU) WRITE(NOT,1000)
 1000 FORMAT (' SUBROUTINE QRDRVR ')
C   PUT A INTO W2
C
      DO 10 I=1,N
      DO 10 J=1,N
   10 W2(I,J) = A(I,J)
C
C   NOW PLACE W2 INTO Q (AS IF FOR EXTENDED PRECISION)
C                          -- --
      DO 20 J=1,N
      DO 20 I=1,N
   20 Q(I,J) = W2(I,J)
      IF(.NOT.LEQU) GO TO 40
      WRITE(NOT,1001)
 1001 FORMAT (' ELEMENTS OF ARRAY Q')
      CALL WRITES(Q,N,N,KR)
   40 CONTINUE
C
C     PUT MATRIX INTO UPPER HESSENBERG FORM
      CALL SUBQIA (Q,N,KR,VI)
      IF(.NOT.LEQU) GO TO 50
      WRITE(NOT,1002)
 1002 FORMAT (' UPPER HESSENBERG FORM OF Q OUT OF SUBQIA IS')
      CALL WRITES(Q,N,N,KR)
   50 CONTINUE
C
C     CALCULATE EIGENVALUES USING QR METHOD
      CALL QRCON  (Q,N,RR,RI,KR,0,0)
C
C     ALIGN EIGENVALUES INTO INCREASING ORDER
C
      NM1 = N-1
      DO 35 J=1,NM1
      W2MIN = RR(J)
      WMIN = RI(J)
      IMIN = J
      JP1 = J+1
      DO 30 I=JP1,N
      IF (W2MIN .LE. RR(I)) GO TO 30
      W2MIN = RR(I)
      WMIN = RI(I)
      IMIN = I
   30 CONTINUE
      RR(IMIN) = RR(J)
      RI(IMIN) = RI(J)
      RI(J) = WMIN
   35 RR(J) = W2MIN
      IF(.NOT.LEQU) GO TO 60
      WRITE(NOT,1004)
 1004 FORMAT (' REAL ROOTS OUT OF QRCON ARE ')                         0 9612
      CALL WRITES(RR,1,N,1)                                            0 9613
      WRITE(NOT,1005)                                                  0 9614
 1005 FORMAT (' IMAGINARY ROOTS OUT OF QRCON ARE ')                    0 9615
      CALL WRITES(RI,1,N,1)                                            0 9616
   60 CONTINUE                                                         0 9617
      RETURN                                                           0 9618
      END                                                              0 9619
```

```
      SUBROUTINE QRCON (A,M,ROOTR,ROOTI,KR,IPRNT,ITIMES)               0 9620
C                                                                      0 9621
C                                                   DEBUG #  85        0 9622
      IMPLICIT REAL*8(A-H,O-Z)                                         0 9623
C       PROGRAM TO CALL QR TRANSFORMATION, MAXIMUM ITER IS 50.         0 9624
      DIMENSION A(KR,1),ROOTR(1),ROOTI(1)                              0 9625
      TEST=10.D0**(-20)*10.D0**(2*ITIMES)                              0 9626
      IF (ITIMES .EQ. 0) TEST=10.D0**(-20)                             0 9627
      N = M                                                            0 9628
      IF(IPRNT)  100, 110, 100                                         0 9629
  100 WRITE (6,104)                                                    0 9630
  110 ZERO = 0.D0                                                      0 9631
      JJ=1                                                             0 9632
  120 XNN=0.D0                                                         0 9633
      XN2=0.D0                                                         0 9634
      AA = 0.D0                                                        0 9635
      B = 0.D0                                                         0 9636
      C = 0.D0                                                         0 9637
      DD = 0.D0                                                        0 9638
      R=0.D0                                                           0 9639
      SIG=0.D0                                                         0 9640
      ITER = 0                                                         0 9641
  130 IF(N-2)     140, 180, 190                                        0 9642
  140 IF(IPRNT)  150, 160, 150                                         0 9643
  150 WRITE (6,105)A(1,1)                                              0 9644
  160 ROOTR(1) = A(1,1)                                                0 9645
      ROOTI(1) = 0.D0                                                  0 9646
  170 RETURN                                                           0 9647
  180 JJ=-1                                                            0 9648
  190 X = (A(N-1,N-1) - A(N,N))**2                                     0 9649
      S = 4.D0*A(N,N-1)*A(N-1,N)                                       0 9650
      ITER = ITER + 1                                                  0 9651
      IF (X .EQ. 0.D0) GO TO 240                                       0 9652
      IF (DABS(S/X) .GT. 1.D-8) GO TO 240                              0 9653
  200 IF(DABS(A(N-1,N-1))-DABS(A(N,N)))  220, 220, 210                 0 9654
  210 E = A(N-1,N-1)                                                   0 9655
      G = A(N,N)                                                       0 9656
      GO TO 230                                                        0 9657
  220 G = A(N-1,N-1)                                                   0 9658
      E = A(N,N)                                                       0 9659
  230 F = 0.D0                                                         0 9660
      H = 0.D0                                                         0 9661
      GO TO 290                                                        0 9662
  240 S = X + S                                                        0 9663
      X = A(N-1,N-1) + A(N,N)                                          0 9664
      IF(S)     260, 250, 250                                          0 9665
```

```
250 SQ=DSQRT(S)                                                     0 9666
    F=0.D0                                                          0 9667
    H=0.D0                                                          C 9668
    IF (X) 260, 260, 270                                            0 9669
260 E=(X-SQ)/2.D0                                                   C 9670
    G=(X+SQ)/2.D0                                                   0 9671
    GO TO 290                                                       C 9672
270 G=(X-SQ)/2.D0                                                   0 9673
    E=(X+SQ)/2.D0                                                   C 9674
    GO TO 290                                                       C 9675
280 F =DSQRT(-S)/2.D0                                               0 9676
    E=X/2.D0                                                        C 9677
    G=E                                                             0 9678
    H=-F                                                            0 9679
290 IF(JJ)  310, 300, 300                                           0 9680
300 D = TEST   *(DABS(G) + F)                                       C 9681
    IF(DABS(A(N-1,N-2)) .GT. D)  GO TO 340                          0 9682
310 IF(IPRNT)  320, 330, 320                                        0 9683
320 WRITE (6,105)E,F, ITER                                          C 9684
    WRITE (6,105)G,H                                                C 9685
330 ROOTR(N) = E                                                    C 9686
    ROOTI(N) = F                                                    C 9687
    ROOTR(N-1) = G                                                  C 9688
    ROOTI(N-1) = H                                                  C 9689
    N=N-2                                                           C 9690
    IF(JJ)  170, 120, 120                                           0 9691
340 IF(DABS(A(N,N-1)) .GT. TEST   *DABS(A(N,N)))  GO TO 380         C 9692
350 IF(IPRNT)  360, 370, 360                                        C 9693
360 WRITE (6,105)A(N,N), ZERO, ITER                                 C 9694
370 ROOTR(N) = A(N,N)                                               0 9695
    ROOTI(N) = 0.D0                                                 0 9696
    N=N-1                                                           0 9697
    GO TO 120                                                       0 9698
380 IF(DABS(DABS(XNN/A(N,N-1))-1.D0)-1.0D-8)  400, 400, 390         0 9699
390 IF(DABS(DABS(XN2/A(N-1,N-2))-1.D0)-1.0D-8)  400, 400, 490       C 9700
400 VQ=DABS(A(N,N-1))-DABS(A(N-1,N-2))                              C 9701
    IF (ITER-15) 520, 410, 440                                      0 9702
410 IF(VQ) 420, 420, 430                                            0 9703
420 R = A(N-1,N-2)**2                                               C 9704
    SIG = 2.D0*A(N-1,N-2)                                           0 9705
    GO TO 570                                                       C 9706
430 R = A(N,N-1)**2                                                 0 9707
    SIG = 2.D0*A(N,N-1)                                             0 9708
    GO TO 570                                                       0 9709
440 IF(VQ)   470, 470, 450                                          C 9710
450 IF(IPRNT)  460, 330, 460                                        0 9711
460 WRITE (6,107)A(N-1,N-2)                                         C 9712
    GO TO 320                                                       0 9713
470 IF(IPRNT)  480, 370, 480                                        0 9714
480 WRITE (6,107)A(N,N-1)                                           C 9715
    GO TO 360                                                       C 9716
490 IF(ITER .GT. 50) GO TO 400                                      C 9717
    IF(ITER .GT. 5 ) GO TO 520                                      C 9718
500 R=E*E+F*F                                                       C 9719
    Z1= (E-AA) *(E-AA)+(F-B)*(F-B)                                  0 9720
    IF (R .NE. 0.D0) GO TO 501                                      0 9721
    Z1 = 0.D0                                                       C 9722
    GO TO 502                                                       C 9723
501 Z1 = Z1/R                                                       C 9724
502 R=G*G+H*H                                                       0 9725
```

```
      ZZ = (B+C)*(B-C)+(A-D)*(A-D)                                          0 9726
      IF (S .LE. 0.0) GO TO 503                                             0 9727
      ZZ = 0.D0                                                             0 9728
      GO TO 504                                                             0 9729
  503 ZZ=ZZ/Z                                                               0 9730
  504 CONTINUE                                                              0 9731
      IF(Z1-0.0D0)  510, 510, 540                                           0 9732
  510 IF(Z2-0.0D0)   520, 520, 530                                          0 9733
  520 R=R*C-E*H                                                             0 9734
      SIG=E*S                                                               0 9735
      GO TO 570                                                             0 9736
  530 R=E*E                                                                 0 9737
      SIG=F*S                                                               0 9738
      GO TO 570                                                             0 9739
  540 IF(Z2-0.0D0) 550, 550, 560                                            0 9740
  550 R=G*C                                                                 0 9741
      SIG=C*S                                                               0 9742
      GO TO 570                                                             0 9743
  560 R = 0.D0                                                              0 9744
      SIG = 0.D0                                                            0 9745
  570 XNN=A(N,N-1)                                                          0 9746
      XN2=A(N-1,N-2)                                                        0 9747
      CALL SH2 (A,N,R,SIG,D,KR)                                             0 9748
      XA=D                                                                  0 9749
      E=R                                                                   0 9750
      C=C                                                                   0 9751
      D=SM                                                                  0 9752
      GO TO 100                                                             0 9753
  104 FORMAT(////1X, 9HREAL PART  6X  14HIMAGINARY PART, 20X                0 9754
     1  13HFINAL AS ZERO 6X 4HITER //)                                      0 9755
  105 FORMAT(1X,D15.8,3X,D15.8, 4X I3)                                      0 9756
  107 FORMAT(30X  D13.8)                                                    0 9757
      END                                                                   0 9758



      SUBROUTINE SUBDIA (A,N,KR,B)                                          0 9759
C                                                            DEBUG #  86    0 9760
      IMPLICIT REAL*8(A-H,O-Z)                                              0 9761
      DIMENSION A(KR,1),B(1)                                                0 9762
C     SUBROUTINE TO PUT MATRIX IN UPPER HESSENBERG FORM.                    0 9763
      IF (N - 2) 200, 200, 100                                              0 9764
  100 DO 250 LC = 3,N                                                       0 9765
      K = N - LC + 3                                                        0 9766
      N1 = N - 1                                                            0 9767
      N2 = N - 2                                                            0 9768
      NI = N1                                                               0 9769
      DIV = DABS(A(N,N-1))                                                  0 9770
      DO 120 J = 1,N2                                                       0 9771
      IF(DABS(A(N,J))- DIV) 120, 120, 110                                   0 9772
  110 NI = J                                                                0 9773
      DIV = DABS(A(N,J))                                                    0 9774
  120 CONTINUE                                                              0 9775
      IF(DIV) 130, 250, 130                                                 0 9776
  130 IF(NI - N1) 140, 170, 140                                             0 9777
  140 DO 150 J = 1,N                                                        0 9778
      DIV = A(J,NI)                                                         0 9779
      A(J,N1) = A(J,N1)                                                 0 9780
  150 A(J,N1) = DIV                                                     0 9781
      DO 160 J = 1,M                                                    0 9782
      DIV = A(N1,J)                                                     0 9783
      A(N1,J) = A(N1,J)                                                 0 9784
  160 A(N1,J) = DIV                                                     0 9785
  170 DO 180 K = 1, N1                                                  0 9786
  180 B(K) = A(N,K)/A(N,N-1)                                            0 9787
      DO 240 J = 1,M                                                    0 9788
      SUM = 0.D0                                                        0 9789
      IF (J - N1) 190, 220, 220                                         0 9790
  190 IF(B(J))  200, 220, 200                                           0 9791
  200 A(N,J) = 0.D0                                                     0 9792
      DO 210 K = 1,N1                                                   0 9793
      A(K,J) = A(K,J) - A(K,N1)*B(J)                                    0 9794
  210 SUM = SUM + A(K,J)*B(K)                                           0 9795
      GO TO 240                                                         0 9796
  220 DO 230 K = 1,N1                                                   0 9797
  230 SUM = SUM + A(K,J)*B(K)                                           0 9798
  240 A(N1,J) = SUM                                                     0 9799
  250 CONTINUE                                                          0 9800
  260 RETURN                                                            0 9801
      END                                                               0 9802
```

```
      SUBROUTINE QR2 (A,N,R,SIG,D,KR)                                   0 9803
C                                                          DEBUG #  87  0 9804
C                SUBROUTINE TO BE USED BY QRCON                         0 9805
      IMPLICIT REAL*8(A-H,O-Z)                                          0 9806
      DIMENSION A(KR,1),G(3),PSI(2)                                     0 9807
      N1 = N - 1                                                        0 9808
      IA = N - 2                                                        0 9809
      IP = IA                                                           0 9810
      IF(N-3) 450, 140, 100                                             0 9811
  100 DO 130 J = 3,N1                                                   0 9812
      J1 = N - J                                                        0 9813
      IF(DABS(A(J1+1,J1))-D)    140, 140, 110                           0 9814
  110 DEN = A(J1+1,J1+1)*(A(J1+1,J1+1)-SIG)+A(J1+1,J1+2)*A(J1+2,J1+1)+R 0 9815
      IF(DEN) 120, 130, 120                                             0 9816
  120 IF(DABS(A(J1+1,J1)*A(J1+2,J1+1)*(DABS(A(J1+1,J1+1)+A(J1+2,J1+2)   0 9817
     1-SIG)+DABS(A(J1+3,J1+2)))/DEN)-D)  140, 140, 130                  0 9818
  130 IP=J1                                                             0 9819
  140 DO  150  J=1,IP                                                   0 9820
      J1=IP-J+1                                                         0 9821
      IF (DABS(A(J1+1,J1))-D)  160, 160, 150                            0 9822
  150 IQ=J1                                                             0 9823
  160 DO  400  I=IP,N1                                                  0 9824
      IF(I-IP)    180, 170, 180                                         0 9825
  170 G(1)=A(IP,IP)*(A(IP,IP)-SIG)+A(IP,IP+1)*A(IP+1,IP)+R              0 9826
      G(2)=A(IP+1,IP)*(A(IP,IP)+A(IP+1,IP+1)-SIG)                       0 9827
      G(3)=A(IP+1,IP)*A(IP+2,IP+1)                                      0 9828
      A(IP+2,IP)=0.D0                                                   0 9829
      GO TO 210                                                         0 9830
  180 G(1)=A(I,I-1)                                                     0 9831
      G(2)=A(I+1,I-1)                                                   0 9832
      IF(I-IA)    190, 190, 200                                         0 9833
```

```
190 G(3)=A(I+2,I-1)
    GO TO 210
200 G(3)=0.D0
210 XK=DSQRT(G(1)*G(1)+G(2)*G(2)+G(3)*G(3))
    IF(G(1).LT.0.D00) XK=-XK
220 IF(XK)    230, 240, 230
230 AL=G(1)/XK+1.D0
    PSI(1)=G(2)/(G(1)+XK)
    PSI(2)=G(3)/(G(1)+XK)
    GO TO 250
240 AL=2.D0
    PSI(1)=0.D0
    PSI(2)=0.D0
250 IF(I-IJ)    260, 290, 260
260 IF(I-I2)     280, 270, 280
270 A(I,I-1)=-A(I,I-1)
    GO TO 290
280 A(I,I-1)=-XK
290 DO 340   J=I,N
    IF(I-IA)  300, 300, 310
300 C=PSI(2)*A(I+2,J)
    GO TO 320
310 C=0.D0
320 E=AL*(A(I,J)+PSI(1)*A(I+1,J)+C)
    A(I,J)=A(I,J)-E
    A(I+1,J)=A(I+1,J)-PSI(1)*E
    IF(I-IA)    330, 330, 340
330 A(I+2,J)=A(I+2,J)-PSI(2)*E
340 CONTINUE
    IF(I-IA)    350, 350, 360
350 L=I+2
    GO TO   370
360 L=N
370 DO   420   J=IJ,L
    IF(I-IA)  380, 380, 390
380 C=PSI(2)*A(J,I+2)
    GO TO   400
390 C=0.D0
400 E=AL*(A(J,I)+PSI(1)*A(J,I+1)+C)
    A(J,I)=A(J,I)-E
    A(J,I+1)=A(J,I+1)-PSI(1)*E
    IF(I-IA)  410, 410, 420
410 A(J,I+2)=A(J,I+2)-PSI(2)*E
420 CONTINUE
    IF(I-N+3)    430, 430, 440
430 E=AL*PSI(2)*A(I+3,I+2)
    A(I+3,I)=-E
    A(I+3,I+1)=-PSI(1)*E
    A(I+3,I+2)=A(I+3,I+2)-PSI(2)*E
440 CONTINUE
450 RETURN
    END




    SUBROUTINE HWRITE (K,HR1,HI1,HR2,HI2,N1,N2,ANAA1,ANAA2)
C
```

```
C                                                        DEBUG #  88
      IMPLICIT REAL*8(A-H,O-Z)
C
C     ---LATEST MOD ON 09/04/75
C     ---LATEST MOD ON 09/04/75
C-----SUBROUTINE PULLS UP NEW PAGE VIA PAGEHD, PRINTS OUT
C      IDENTIFICATION(S), ANAM1 AND ANAM2, THEN PRINTS ROOTS.
C
C             -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C                     ALL ARGUMENTS ARE INPUT
C
C  K       = NO. OF ROOT SETS TO PRINT.
C  RR1     = REAL ROOTS (FIRST SET)
C  RI1     = IMAG ROOTS (FIRST SET)
C  RR2     = REAL ROOTS (SECOND SET)
C  RI2     = IMAG ROOTS (SECOND SET)
C  N1      = ROOT COUNT (FIRST SET)
C  N2      = ROOT COUNT (SECOND SET)
C
C  ANAM1   = 4 CHARACTER ALPHANUMERIC TITLE (FIRST SET)
C  ANAM2   = 4 CHARACTER ALPHANUMERIC TITLE (SECOND SET)
C
      DIMENSION RR1(1), RI1(1), RR2(1), RI2(1)
      DATA NOT /
     1       6 /
C
  101 FORMAT (///20X,A4,34X,A4,//5X,2HNO,5X,9HREAL PART ,
     1         3X,14HIMAGINARY PART,11X,9HREAL PART,3X,14HIMAGINARY PART,
     2         //)
  102 FORMAT (5X,I2,3X,D12.5,5X,D12.5,9X,D12.5,3X,D12.5)
  103 FORMAT (5X,I2,41X,D12.5,5X,D12.5)
C
      CALL PAGEHD
      IF (K .EQ. 2) GO TO 20
      WRITE (NOT,101) ANAM1
      IL = 0
      DO 10 I=1,N1
      IL = IL+1
      IF (IL .LE. 45) GO TO 10
      IL = 0
      CALL PAGEHD
      WRITE (NOT,101) ANAM1
   10 WRITE (NOT,102) I,RR1(I), RI1(I)
      RETURN
   20 CONTINUE
C
      L = MAX0(N1,N2)
      WRITE (NOT,101) ANAM1, ANAM2
      IL = 0
      DO 40 I=1,L
      IL=IL+1
      IF (IL .LE. 45) GO TO 25
      IL=0
      CALL PAGEHD
      WRITE (NOT,101) ANAM1, ANAM2
   25 CONTINUE
      IF (I .GT. N1  .OR.  I .GT. N2) GO TO 30
      WRITE (NOT,102) I, RR1(I),RI1(I),RR2(I),RI2(I)
      GO TO 40
   30 CONTINUE
      IF (I .GT. N2) WRITE (NOUT,102) I, NR1(I), R11(I)
      IF (I .GT. N1) WRITE (NOUT,103) I, NR2(I), R12(I)
   40 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SPLOT (TITLE,FMAX,FMIN,DBMIN,DBMAX)
C                                                    DFLOG #  89
C
      REAL*8 IRUNNO, IDATE
      REAL*8 UNAME, TITLE1, TITLE2
C  SUBROUTINE PLOTS FREQUENCY DATA ON SEMILOG GRID.
C
C                -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C  TITLE   = INPUT 80 CHARACTER ALPHANUMERIC TITLE.
C  FMAX    = INPUT UPPER RANGE LIMIT FOR FREQUENCY SWEEP.
C  FMIN    = INPUT LOWER RANGE LIMIT FOR FREQUENCY SWEEP.
C  DBMIN   = MINIMUM DB TO PLOT.
C  DBMAX   = MAXIMUM DB TO PLOT.
C
      DIMENSION TITLE(1),DBTITL(36),FTITLE(7),ANUM(9)
      DIMENSION ZX(2), ZY(2)
      DIMENSION FAZE(500),DBAMP(500)
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE
      COMMON /LSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)
      COMMON /LEMODE/ AMODE(230)
      COMMON /ESTORE/
     *              SAVED(500), SAVEP(500), SAVFD(500), SAVEA(500), KSAVE
C
      DATA DBTITL /
     *   4H-200,4H-190,4H-180,4H-170,4H-160,4H-150,4H-140,4H-130,4H-120,
     *   4H-110,4H-100,4H-90.,4H-80.,4H-70.,4H-60.,4H-50.,4H-40.,4H-30.,
     *   4H-20.,4H-10.,4H 0.0,4H 10.,4H 20.,4H 30.,4H 40.,4H 50.,4H 60.,
     *   4H 70.,4H 80.,4H 90.,4H100.,4H110.,4H120.,4H130.,4H140.,4H150./
C
      DATA FTITLE /
     * 4HE-2 ,4HE-1 ,4HE00 ,4HE01 ,4HE02 ,4HE03 ,4HE04 /
C
      DATA ANUM /
     *   1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9 /
C
      CALL RSETMG (AMODE)
C
C-----ESTABLISH INITIAL DECADE FACTOR
      AG = ALOG10(FMIN)
      IF ((AG.LT.0.) .AND. (AG-FLOAT(IFIX(AG)).NE. 0.)) AG = AG-1.
      FACTOR = 10.**IFIX(AG)
      FMIN = FACTOR
C
C  DETERMINE GRID FREQUENCY
      AG = ALOG10(FMAX)
      IF(AG .GE. 0.0 .AND. (AG-FLOAT(IFIX(AG))) .NE. 0.0)
     1   AG = AG + 1.0
```

```
      K = ALOG10(FMIN)
      K = K-1
C
      GP = 10.**IFIX(AG)
      NCC = IFIX(AG) - IFIX(ALOG10(FMIN))
C
      CALL SCALE (DBMAX,DBMIN,10,330,DBL,ZIP)
      IF (DBL .LT. -200.) DBL = -200.
C
C--CALCULATE LABEL LOCATORS.
C
      ISY = (DBL + 200.)/10. + 1.
      ISX = ALOG10(FMIN) + 3.0
C
C
C     MODIFY PHASE VALUES TO PLOT USING DB MAPPING SCALE.
      R1 = (DBU-DBL)* 0.0025
      R2 = (DBU+DBL)* 0.5
C
      DO 1 I=1,KSAVE
      YDB = SAVED(I)
      IF (YDB .LT. DBL) YDB = DBL
      IF (YDB .GT. DBU) YDB = DBU
      DSAVE(I) = YDB
      XFAZE = SAVEP(I)
      IF (XFAZE .GT. 180.) XFAZE = XFAZE - 360.
    1 FAZE(I) = R1*XFAZE + R2
C
C
C  GENERATE PLOT GRID WITH SCALES, LABELS, AND TITLE.
C
C  INDICATE UTILIZATION OF NORMALIZED SPACE
C
      CALL SETSMG (AMODE,19,1.)
      CALL SETSMG (AMODE,20,1.)
C
C
C  USE FULL SPACES FOR PUTTING ON RIGHT HAND LABELS.
C
      CALL OBJCTG (AMODE,0.,0.,1.,1.)
      CALL SUBJEG (AMODE,0.,0.,1.,1.)
C--BOX THE GRID.
      CALL BOXG (AMODE,0.0,0.0,0.0,1.0,0.0,0.0,0.99,1.0,0.99)
C--INCREASE CHARACTER SIZE
      CALL SETSMG (AMODE,45,1.5)
C--PUT ON THE MODE TITLE.
      CALL LEGNDG (AMODE,0.45,0.93,4,4HMODE)
C
C--SET FLAG TO ORIENT LABEL AT 90. DEGREES.
C
      CALL SETSMG (AMODE,46,90.)
C
C--PUT LABEL ON RHS.
C
      CALL LEGNDG (AMODE,0.960,0.4355,11,11HPHASE - DEG)
C--LABEL THE Y-AXIS.
      CALL LEGNDG (AMODE,0.06,0.44,9,9HGAIN - DB )
C
C--RESET CHARACTER SIZE
      CALL SETSMG (AMODE,45,1.0)
```

```
C
C--RESET ORIENTATION OF LABEL.
       CALL SETSMG (AMODE,40,0.)
C
C--PLACE SCALE ON RHS. (USING LEGNDG)
C
       CALL LEGNDG (AMODE,.907,.910,4,4H 200)
       CALL LEGNDG (AMODE,.907,.832,4,4H 160)
       CALL LEGNDG (AMODE,.907,.754,4,4H 120)
       CALL LEGNDG (AMODE,.907,.676,4,4H  80)
       CALL LEGNDG (AMODE,.907,.598,4,4H  40)
       CALL LEGNDG (AMODE,.907,.520,4,4H   0)
       CALL LEGNDG (AMODE,.907,.442,4,4H -40)
       CALL LEGNDG (AMODE,.907,.364,4,4H -80)
       CALL LEGNDG (AMODE,.907,.286,4,4H-120)
       CALL LEGNDG (AMODE,.907,.208,4,4H-160)
       CALL LEGNDG (AMODE,.907,.130,4,4H-200)
C
C--PUT ON ADDITIONAL TITLES.
       CALL SETSMG (AMODE,45,1.25)
       CALL LINESG (AMODE,0.,0.05,0.02)
       CALL TEXTG (AMODE,8,IRUNNO)
       CALL TEXTG (AMODE,8,IDATE)
       CALL LINESG (AMODE,0.,0.,0.02)
       CALL TEXTG (AMODE,8,UNAME(1))
       CALL TEXTG (AMODE,8,UNAME(2))
       CALL TEXTG (AMODE,8,UNAME(3))
       CALL SETSMG (AMODE,45,1.)
C
C  ESTABLISH SUBJECT AND OBJECT SPACES.
C
       CALL OBJCTG  (AMODE,0.11,0.13,0.89,0.91)
       CALL SUBJEG  (AMODE,FMIN,DUL,DP,DMU)
C
C  INDICATE NONLINEAR X-SCALE.
C
       CALL SETSMG (AMODE,23,1.)
C
C  INDICATE DESIRED EMPHASIS ON Y+D ONLY.
C
       CALL SETSMG (AMODE,100,2.)
C
C--PUT ON GRID.
C
       CALL GRIDG (AMODE,1,,3,3,1,2)
C
C--PUT ON TITLE AND LABELS
C
       CALL SETSMG (AMODE,40,1.5)
       CALL TITLEG (AMODE,15,18HFREQUENCY - RAD/SEC,
      1                          0,DUMY,
      2                         00,TITLE)
       CALL SETSMG (AMODE,45,1.)
C
C--LABEL Y-AXIS.
C
       CALL LABELG (AMODE,1,10,4,UOTITL(ISY))
C
C--LABEL X-AXIS)
C
```

```
      CALL LABELG (AMODE,0, 1.,4,FTITLE(ISX))
C
C--SET FLAG TO ORIENT LABEL AT 90 DEGREES.
C
      CALL SETSMG (AMODE,46,90.)
C
C--PUT ON LOG GRID AND INTERMEDIATE LABELS.
C
      ZYI(1) = DBL
      ZYI(2) = DBU
      DO 7 J=1,NLC
      K = K+1
      DO 7 I=2,9
      XI = FLOAT(I) * 10. **K
      ZXI(1) = XI
      ZXI(2) = XI
      CALL LINESG (AMODE,2,ZXI,ZYI)
C
      CALL OBJCTG (AMODE,0,11,0.,0.89,1.)
      CALL SUBJEG (AMODE,FMIN,0.,UP,1.)
      CALL LEGNDG (AMODE,XI,0,12,1,ANUM(I))
      CALL OBJCTG (AMODE,0,11,0.13,0.89,0.91)
      CALL SUBJEG (AMODE,FMIN,DBL,UP,DBU)
C
    7 CONTINUE
C
C--RESET ORIENTATION OF LABEL.
C
      CALL SETSMG (AMODE,46,0.)
C
C--PUT DATA ON PLOT.
C
      CALL LINESG (AMODE,KSAVE,SAVED,FAZE)
      CALL TEXTG  (AMODE,1,IHP)
      CALL LINESG (AMODE,KSAVE,SAVED,DBAMP)
      CALL TEXTG  (AMODE,1,IMG)
      CALL PAGEG  (AMODE,0,0,1)
      RETURN
      END




      SUBROUTINE NIPLOT (TITLE,DBMIN,DBMAX)
C                                                              DEBUG #  90
      REAL*8 IRUNNO, IDATE
      REAL*8 UNAME, TITLE1, TITLE2
C
C-----SUBROUTINE DISPLAYS S-PLANE FREQUENCY
C-----              VIA THE NICHOLS PLOT
C
C-----            -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C-----TITLE  = INPUT 30 CHARACTER ALPHANUMERIC TITLE.
C-----DBMIN  = INPUT LOWER DBLIMIT.
C-----DBMAX  = UPPER DBLIMIT.
C
      DIMENSION TITLE(1), DBTITL(30)
      COMMON /ZCBAGE/ AMGDE(200)                                        010176
C                                                                       010177
      COMMON /ZCSTART/ IRUNNO, IDATE, NPAGE                             010178
      COMMON /ZCSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)                 010179
      COMMON /ZPSTORE/                                                  010180
     *               SAVEX(500), SAVEY(500), SAVEO(500), SAVEA(500), KSAVE 010181
C                                                                       010182
      DATA OTITLE /                                                     010183
     *   4H-200,4H-190,4H-180,4H-170,4H-160,4H-150,4H-140,4H-130,4H-120,  010184
     *   4H-110,4H-100,4H-90.,4H-80.,4H-70.,4H-60.,4H-50.,4H-40.,4H-30.,  010185
     *   4H-20.,4H-10.,4H 0.0,4H 10.,4H 20.,4H 30.,4H 40.,4H 50.,4H 60.,  010186
     *   4H 70.,4H 80.,4H 90.,4H100.,4H110.,4H120.,4H130.,4H140.,4H150./  010187
C                                                                       010188
      CALL RESETG (AMGDE)                                               010189
C                                                                       010190
      CALL SETSMG (AMGDE,1.,1.)                                         010191
      CALL SETSMG (AMGDE,2.,1.)                                         010192
C                                                                       010193
C-----SET FULL SPACE FOR BOX AND TITLING.                               010194
      CALL OBJCTG (AMGDE,0.,0.,1.,1.)                                   010195
      CALL SBJCTG (AMGDE,0.,0.,1.,1.)                                   010196
C-----BOX THE GRID.                                                     010197
      CALL BOXG (AMGDE,0.,0.,0.,0.,1.,0.,0.,0.,0.,999.,1.,0.,0.,9)      010198
C-----INCREASE CHARACTER SIZE.                                          010199
      CALL SETSMG (AMGDE,45,1.5)                                        010200
C-----SET ORIENTATION AT 90 DEGREES.                                    010201
      CALL SETSMG (AMGDE,46,90.)                                        010202
C-----LABEL Y-AXIS.                                                     010203
      CALL LEGNDG (AMGDE,0.06,0.35,26,26HAMPLITUDE RATIO (DECIBELS))    010204
C-----RESET ORIENTATION ANGLE.                                          010205
      CALL SETSMG (AMGDE,46,0.)                                         010206
C-----LABEL X-AXIS.                                                     010207
      CALL LEGNDG (AMGDE,0.4,0.06,15,15HPHASE (DEGREES))                010208
C-----PUT ON TITLE.                                                     010209
      CALL LEGNDG (AMGDE,0.2,0.94,60,TITLE1)                            010210
      CALL LEGNDG (AMGDE,0.4,0.90,7,7HNICHOLS)                          010211
C-----PUT ON ADDITIONAL TITLING.                                        010212
      CALL SETSMG (AMGDE,45,1.25)                                       010213
      CALL LINESG (AMGDE,0.,0.05,0.02)                                  010214
      CALL TEXTG (AMGDE,6,IRUNNO)                                       010215
      CALL TEXTG (AMGDE,6,IDATE)                                        010216
      CALL LINESG (AMGDE,0.,0.5,0.32)                                   010217
      CALL TEXTG (AMGDE,6,UNAME(1))                                     010218
      CALL TEXTG (AMGDE,6,UNAME(2))                                     010219
      CALL TEXTG (AMGDE,6,UNAME(3))                                     010220
      CALL SETSMG (AMGDE,45,1.)                                         010221
C                                                                       010222
C-----SET LIMITS FOR OBJECT SPACE.                                      010223
C                                                                       010224
      X1 = 0.1                                                          010225
      X2 = 0.98                                                         010226
      Y1 = 0.12                                                         010227
      Y2 = 0.9                                                          010228
C                                                                       010229
      CALL OBJCTG (AMGDE,X1,Y1,X2,Y2)                                   010230
C                                                                       010231
C-----ESTABLISH SUBJECT SPACE.                                          010232
C                                                                       010233
      ORL = SAVEO(1)                                                    010234
      DO 20 I=2,KSAVE                                                   010235
```

```
   20 DBL = AMIN1 (DBL,SAVED(I))
      IF (DBL .LT. DBMIN) DBL = DBMIN
      DBU = DBL + 100.
C
      ISY = (DBL + 200.)/10. + 1.
      CALL SUBJEG (AMODE,-140.,DBL ,260.,DBU)
C
      CALL GRIDG (AMODE,10.,5.,4,2)
C
C-----NOW HAVE CARTESIAN GRID.
C-----PUT ON TITLE AND LABELS.
C
C
      CALL LABELG (AMODE,0,40.,3,
     1         33H240280320 0  40 80120160200240280)
C
      CALL LABELG (AMODE,1,10.,4,DBTITL(ISY))
C
      KOPH = 0
      POPHI= 0.0
C-----NOW HAVE GRID,TITLE, AND LABELS.
C
      DO 1480 I=1,KSAVE
C
      DPHI = SAVEP(I)
      DBELL= SAVED(I)
C
      IF (I .GT. 1) GO TO 1400
C
C-----FIRST POINT
C
      IF (DPHI .GT. 225.) DPHI = DPHI - 360.
      IF (DBELL .LT. DBL  .OR. DBELL .GT. DBU) GO TO 1425
C
C-----POSITION BEAM (AND WRITE FREQUENCY ON GRAPH)
C
      CALL LINESG (AMODE,0,DPHI,DBELL)
      GO TO 1480
C
 1400 IF (DPHI .GT. 225.) DPHI = DPHI - 360.
 1420 IF (DBELL .GT. DBL   .AND.  DBELL .LT. DBU) GO TO 1430
      IF (KOPH .NE. 0)  GO TO 1480
 1425 KOPH = 1
      GO TO 1480
 1430 IF (KOPH .EQ. 0) GO TO 1440
      KOPH = 0
      GO TO 1450
 1440 IF (ABS(POPHI-DPHI) .LT. 330.) GO TO 1470
 1450 CALL LINESG (AMODE,0,DPHI,DBELL)
      GO TO 1480
 1470 CALL LINESG (AMODE,1,DPHI,DBELL)
 1480 POPHI = DPHI
      CALL RSETMG (AMODE)
      CALL PAGEG (AMODE,0,0,1)
C
      RETURN
      END
      SUBROUTINE NYPLOT (TITLE,AMIN,AMAX)                               010293
C                                                        DEBUG #  91    010294
      REAL*8 IRUNNO, IDATE                                              010295
      REAL*8 UNAME, TITLE1, TITLE2                                      010296
C                                                                       010297
C-----SUBROUTINE PLOTS AR VS. DPHI ON A POLAR GRID.                     010298
C                                                                       010299
C                                                                       010300
C               -----SUBROUTINE ARGUMENT DESCRIPTIONS-----              010301
C                                                                       010302
C  TITLE  = INPUT ALPHANUMERIC TITLE.                                   010303
C                                                                       010304
      DIMENSION TITLE(1)                                                010305
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                              010306
      COMMON /LSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)                  010307
      COMMON /PSTUFF/                                                   010308
     *          SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE 44710309
      COMMON /LZMODE/ AMODE(200)                                        010310
C                                                                       010311
      CALL RSETMG (AMODE)                                               010312
C-----USE FULL SPACE FOR BOX AND TITLING.                               010313
      CALL SETSMG (AMODE,19,1.)                                         010314
      CALL SETSMG (AMODE,20,1.)                                         010315
      CALL OBJCTG (AMODE,0.,0.,1.,1.)                                   010316
      CALL SUBJEG (AMODE,0.,0.,1.,1.)                                   010317
C-----BOX THE GRID.                                                     010318
      CALL BOXG (AMODE,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.99,1.0,0.99)       010319
C-----INCREASE CHARACTER SIZE.                                          010320
      CALL SETSMG (AMODE,45,1.5)                                        010321
C-----PUT ON NYQUIST TITLE.                                             010322
      CALL LEGNDG (AMODE,0.42,0.96,13,13HNYQUIST (DEG))                 010323
C-----PUT ON TITLE.                                                     010324
C                                                                       010325
      CALL LEGNDG (AMODE,0.2,0.94,63,TITLE)                             010326
C-----TITLE X-AXIS.                                                     010327
C                                                                       010328
      CALL LEGNDG (AMODE,0.4,0.08,15,15HPHASE (DEGREES))                010329
C-----TITLE Y-AXIS.                                                     010330
C                                                                       010331
      CALL SETSMG (AMODE,46,90.)                                        010332
      CALL LEGNDG (AMODE,0.06,0.4,9,9HAMPLITUDE)                        010333
      CALL SETSMG (AMODE,46,0.0)                                        010334
C                                                                       010335
C-----PUT ON ADDITIONAL TITLING.                                        010336
      CALL SETSMG (AMODE,45,1.25)                                       010337
      CALL LINESG (AMODE,0.0,0.05,0.02)                                 010338
      CALL TEXTG (AMODE ,8,IRUNNO)                                      010339
      CALL TEXTG (AMODE,8,IDATE)                                        010340
      CALL LINESG (AMODE,0.0,0.3,0.02)                                  010341
      CALL TEXTG (AMODE,8,UNAME(1))                                     010342
      CALL TEXTG (AMODE,8,UNAME(2))                                     010343
      CALL TEXTG (AMODE,8,UNAME(3))                                     010344
      CALL SETSMG (AMODE,45,1.)                                         010345
      CALL RSETMG (AMODE)                                               010346
C                                                                       010347
C-----ESTABLISH LIMITS.                                                 010348
      CALL PLOTSS (AMAX,AMIN,RMAX,RMIN)                                 010349
      IF (RMIN .LT. 0.) RMIN = 0.                                       010350
C                                                                       010351
C                                                                       010352
```

```
      IF (RMAX .EQ. RMIN) RMAX = RMIN+10.
C
      TMAX = 360.
      TMIN = 0.0
C
C-----SET OBJECT SPACE.
      CALL SETSMG (AMODE,19,1.33333)
      CALL SETSMG (AMODE,20,1.)
      CALL OBJCTG (AMODE,0.0,0.25,1.0,0.95)
      CALL PSUBJG (AMODE,RMIN,TMIN,RMAX,TMAX)
C
C-----PUT ON GRID LINES.
C
      CALL SETSMG (AMODE,14,0.00)
      CALL SETSMG (AMODE,45,1.00)
      CALL PSETG (AMODE,0,DO,DI,IDTH,IITH,DLO,DLI,FMTO,FMTI)
C
      DO 10 I=100,109
   10 AMODE(I) = 0.0
C
      CALL PGRIDG (AMODE,DO,DI,IDTH,IITH)
      CALL PLABLG (AMODE,0,DLO,0,FMTO)
      CALL PLABLG (AMODE,1,DLI,0,FMTI)
C
      DO 80 I=1,NSAVE
      R = SAVEA(I)
      T = SAVEP(I)
      IF (I .GT. 1) GO TO 19
C
C-----FIRST POINT
C
      IF (R .LT. RMIN .OR. R .GT. RMAX) GO TO 25
C
C-----POSITION BEAM
      CALL PLINEG (AMODE,0,R,T)
      GO TO 80
   19 CONTINUE
   20 IF (R .GT. RMIN .AND. R .LT. RMAX) GO TO 30
      IF (KOPH .NE. 0) GO TO 80
   25 KOPH = 1
      GO TO 80
   30 IF (KOPH .EQ. 0) GO TO 70
      KOPH = 0
      GO TO 50
   50 CALL PLINEG (AMODE,0,R,T)
      GO TO 80
   70 CALL PLINEG (AMODE,1,R,T)
   80 CONTINUE
C
      CALL PAGEG (AMODE,0,1,1)
      CALL ROUTMG (AMODE)
C
      RETURN
      END
      SUBROUTINE RLPLOT (TITLE,ISNIM,ICYC,IREC)                        010407
C                                                          DEBUG #  92 010408
      REAL*8 IRUNNO, IDATE                                             010409
      REAL*8 UNAME, TITLE1, TITLE2                                     010410
C                                                                      010411
C-----SUBROUTINE PLOTS ROOT LOCI FOR A SINGLE ROOT.                    010412
C                                                                      010413
C               -----SUBROUTINE ARGUMENT DESCRIPTIONS-----             010414
C                                                                      010415
C  TITLE  = INPUT ALPHA NUMERIC TITLE                                  010416
C  ISNIM  = INPUT ROOT IDENTIFICATION PARAMETER.                       010417
C         =1   STARTING POINT IS OPEN LOOP ZERO.                       010418
C         =2   STARTING POINT IS OPEN LOOP POLE.                       010419
C  ICYC   = INPUT TRANSFER FUNCTION COUNT (PLACED ON PLOT).            010420
C  IREC   = INPUT ROOT LOCUS CYCLE NUMBER (PUT ON PLOT)                010421
C                                                                      010422
C                                                                      010423
      DIMENSION TITLE(1), X(500), Y(500)                             47210424
      DIMENSION IGR (99)                                               010425
      COMMON /SD4MDE/ AMODE(230)                                       010426
      COMMON /RSTORE/                                                  010427
     *             SAVE1(500), SAVE2(500), SAVE3(500), SAVE4(500), KSAVE 44710428
      COMMON /DSTART/ IRUNNO, IDATE, NPAGE                             010429
      COMMON /DSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)                 010430
C                                                                      010431
      EQUIVALENCE (SAVE1(1),X(1)), (SAVE2(1),Y(1))                     010432
      DATA IGR /                                                       010433
     1          2H 1,2H 2,2H 3,2H 4,2H 5,2H 6,2H 7,2H 8,2H 9,2H10,     010434
     2          2H11,2H12,2H13,2H14,2H15,2H16,2H17,2H18,2H19,2H20,     010435
     3          2H21,2H22,2H23,2H24,2H25,2H26,2H27,2H28,2H29,2H30,     010436
     4          2H31,2H32,2H33,2H34,2H35,2H36,2H37,2H38,2H39,2H40,     010437
     5          2H41,2H42,2H43,2H44,2H45,2H46,2H47,2H48,2H49,2H50,     010438
     6          2H51,2H52,2H53,2H54,2H55,2H56,2H57,2H58,2H59,2H60,     010439
     7          2H61,2H62,2H63,2H64,2H65,2H66,2H67,2H68,2H69,2H70,     010440
     8          2H71,2H72,2H73,2H74,2H75,2H76,2H77,2H78,2H79,2H80,     010441
     9          2H81,2H82,2H83,2H84,2H85,2H86,2H87,2H88,2H89,2H90,     010442
     A          2H91,2H92,2H93,2H94,2H95,2H96,2H97,2H98,2H99 /         010443
C                                                                      010444
      KSAVE = KSAVE + 1                                                010445
C                                                                      010446
      CALL RSETMG (AMODE)                                              010447
C-----FOR CHIO AND PUT ON EXTRA TITLING USING FULL SPACE.              010448
      CALL SETSMG (AMODE,19,1.)                                        010449
      CALL SETSMG (AMODE,20,1.)                                        010450
      CALL OBJCTG (AMODE,0.,0.,1.,1.)                                  010451
      CALL SUBJCG (AMODE,0.,0.,1.,1.)                                  010452
      CALL BCKG   (AMODE,0,0,0.,0.,1.,0.,0.,.99,1.,0.,##)              010453
C                                                                      010454
C-----INCREASE CHARACTER SIZE                                          010455
      CALL SETSMG (AMODE,45,1.5)                                       010456
      CALL LEGNDG (AMODE,0.,45,0.94,13,13HROOT LOCUS)                  010457
C                                                                      010458
C-----ADDITIONAL TITLING                                               010459
      CALL SETSMG (AMODE,45,1.25)                                      010460
      CALL LINESG (AMODE,0.,0.95,0.02)                                 010461
      CALL TEXTG  (AMODE,8,IRUNNO)                                     010462
      CALL TEXTG  (AMODE,8,IDATE)                                      010463
      CALL LINESG (AMODE,0.,0.3,0.02)                                  010464
      CALL TEXTG  (AMODE,8,UNAME(1))                                   010465
      CALL TEXTG  (AMODE,8,UNAME(2))                                   010466
      CALL TEXTG  (AMODE,6,JNAME(J))
      CALL LINESG (AMODE,0,0.8,0.06)
      CALL TEXTG (AMODE,7,7HICYC = )
      CALL TEXTG (AMODE,2,I3R(ICYC))
      CALL LINESG (AMODE,0,0.8,0.02)
      CALL TEXTG (AMODE,7,7HIREC = )
      CALL TEXTG (AMODE,2,I3R(IREC))
C
C-----TITLE IMAG AXIS ON RHS.
C     ORIENTATE LABEL AT 90 DEGREES.
      CALL SETSMG (AMODE,46,90.)
      CALL LEGNDG (AMODE,0.96,0.42,14,14HIMAGINARY PART)
      CALL SETSMG (AMODE,46,0.)
C
C     ESTABLISH LIMITS ON DATA.
C
      XMIN = X(1)
      XMAX = XMIN
      YMIN = Y(1)
      YMAX = YMIN
C
      DO 10 I=2,NSAVE
      XMIN = AMIN1(XMIN,X(I))
      XMAX = AMAX1(XMAX,X(I))
      YMIN = AMIN1(YMIN,Y(I))
   10 YMAX = AMAX1(YMAX,Y(I))
C
C     SET UP SUBJECT SPACE.
      CALL FACTSS (XMAX,XMIN,XTOP,XFCT)
      CALL FACTSS (YMAX,YMIN,YTOP,YFCT)
      CALL SUBJEG (AMODE,XBOT,YBOT,XTOP,YTOP)
C
C--COMPUTE ARGUMENTS TO USE IN GRIDG AND LABELG.
C
      CALL SETUPG (AMODE,1,JX,JY,IXTH,JYTH,DLX,DLY,XFMT,YFMT)
C
C--RESET GRID MARGINS TO 0.
C
      DO 60 I=106,109
   60 AMODE(I) = 0.
C
C--DRAW THE GRID.
C
      CALL GRIDG (AMODE,JX,JY,IXTH,JYTH)
C
C--NOW LABEL IT.
C
      CALL LABELG (AMODE,0,DLX,0,XFMT)
      CALL LABELG (AMODE,1,DLY,0,YFMT)
C
C--NEXT, TITLE IT.
C
      CALL TITLEG (AMODE,9,9HREAL PART,
     1                      0,JOBTLE,
     2                     60,TITLE)
C
C-----USE TEXTG TO INDICATE STARTING POINT.
      CALL LINESG (AMODE,0,X(1),Y(1))
      IF (ISHIM .EQ. 1) CALL TEXTG (AMODE,1,1HO)
      IF (ISHIM .EQ. 2) CALL TEXTG (AMODE,1,1HX)
```

```
C
C------PLOT THE LOCI.
C
      CALL LINESG (AMODE,KSAVE,X,Y)
      CALL LINESG (AMODE,KSAVE,X,Y)
C
C------PAGE THE FRAME.
C
      CALL PAGEG (AMODE,C,0,1)
      RETURN
      END
```

```
      SUBROUTINE SCALE(AMAX,AMIN,NSG,AU,AL,S)
C                                                    DEBUG #   93
C
C  AMAX=MAX VALUE TO BE PLOTTED
C  AMIN=MIN VALUE TO BE PLOTTED
C  NSG=NUMBER OF MAJOR GRIDS = 10 GENERALLY
C  AU=COMPUTED UPPER SCALE LIMIT
C  AL=COMPUTED LOWER SCALE LIMIT
C  S=SCALE FACTOR PER MAJOR GRID
C
      FNSG=NSG
      AL=0.
      IF(AMAX.GT.AMIN) GO TO 13
      AMIN=AMIN-5.
      AMAX=AMAX+5.
   13 ADIF=AMAX-AMIN
      DO 1 I=1,66
      DO 1 J=1,3
      T=2.**(J-2)*FNSG*.1
      IF(ADIF.LE.T*10.**(I-33)) GO TO 2
    1 CONTINUE
    2 S=T*10.**(I-33)/FNSG
      IF(AMIN) 3,7,4
    3 AL=AL-S
      IF(AL-AMIN) 7,7,3
    4 AL=AL+S
      IF(AL-AMIN) 4,7,6
    6 AL=AL-S
    7 AU=AL+S*FNSG
      IF(AU.GE.AMAX) GO TO 11
      IF(J.LT.3) GO TO 10
      J=0
      I=I+1
   10 J=J+1
      T=2.**(J-2)*FNSG*.1
      AL=0.
      GO TO 2
   11 AL=AL-10.*S
      RETURN
      END
      SUBROUTINE PLOTSS (YMAXIN,YMININ,YTOP,YBOT)
C                                                       DEBUG #  94
C
C  SUBROUTINE SELECTS PLOT UPPER AND LOWER LIMINT FOR A 10 SQUARE
C  LINEAR PLOT GRID.
C
C               -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C  YMAXIN = INPUT  MAXIMUM VALUE TO BE PLOTTED.
C  YMININ = INPUT  MINIMUM VALUE TO BE PLOTTED.
C  YTOP   = OUTPUT UPPER GRID LIMIT.
C  YBOT   = OUTPUT LOWER GRID LIMIT.
C
      DATA NOUT/ 6 /
      YMAX = YMAXIN
      YMIN = YMININ
                                                        NERROR = 1
      IF (YMAX .LT. YMIN) GO TO 999
      IF (YMAX .GT. YMIN) GO TO 21
   11 IF (YMAX .LT. 0.00) GO TO 13
      YMAX = 1.001*YMAX
      YMIN = 0.999*YMIN
      GO TO 15
   13 YMAX = 0.999*YMAX
      YMIN = 1.001*YMIN
   15 IF (YMAX .NE. 0.0) GO TO 21
      YMAX = +0.0
      YMIN = -0.0
C
   21 VALUE = (YMAX-YMIN)/10.
      IF (VALUE .LT. ABS(YMIN/100000.)) GO TO 11
      DO 22 I=1,66
      DO 22 J=1,3
      SCALE = 2.**(J-2) * 10. **(I-33)
   22 IF (SCALE .GE. VALUE) GO TO 31
                                                        NERROR = 2
      GO TO 999
C
   31 NSTEPS = YMIN/SCALE
      YBOT = FLOAT(NSTEPS)*SCALE
   32 IF (YMIN) 34,38,36
   33 YBOT = YBOT-SCALE
   34 IF (YBOT .LE. YMIN) GO TO 38
      GO TO 33
   35 YBOT = YBOT+SCALE
   36 IF (YBOT-YMIN) 35,38,37
   37 YBOT = YBOT-SCALE
   38 YTOP = YBOT+10.*SCALE
      IF (YTOP .GE. YMAX) RETURN
      IF (J .LT. 3) GO TO 39
      J = 0
      I = I+1
   39 J = J+1
      SCALE = 2.**(J-2) * 10.**(I-33)
      GO TO 32
C
  999 WRITE (NOUT,2001) NERROR
 2001 FORMAT (5X,49HERROR ENCOUNTERED IN SUBROUTINE PLOTSS, NERROR = I3,
     *         /,5X, 16HPROGRAM STOPPED.  )
      STOP
      END
      SUBROUTINE ALPHAA (ALPHA,A,Z,NR,NC,KR)                          010639
C                                                    DEBUG #  55      010640
      IMPLICIT REAL*8(A-H,O-Z)                                        010641
      DIMENSION A(KR,1), Z(KR,1)                                      010642
C                                                                     010643
C   CALC. ALPHA TIMES MATRIX A. (ALPHA * A = Z).                      010644
C  MATRICES A,Z MAY SHARE SAME CORE LOCATIONS.                        010645
C  CODED BY  RE WOHLEN.  FEBRUARY 1969.                               010646
C                                                                     010647
C      SUBROUTINE ARGUMENTS                                           010648
C  ALPHA = INPUT  SCALAR.                                             010649
C  A     = INPUT  MATRIX. SIZE(NR,NC).                                010650
C  Z     = OUTPUT RESULT MATRIX. SIZE(NR,NC).                         010651
C  NR    = INPUT  NUMBER OF ROWS IN MATRICES A,Z.                     010652
C  NC    = INPUT  NUMBER OF COLS IN MATRICES A,Z.                     010653
C  KR    = INPUT  ROW DIMENSION OF A,Z IN CALLING PROGRAM.            010654
C                                                                     010655
      DO 10 I=1,NR                                                    010656
      DO 10 J=1,NC                                                    010657
   10 Z(I,J) = ALPHA * A(I,J)                                         010658
      RETURN                                                          010659
      END                                                             010660
```

```
      SUBROUTINE BABT (A,B,Z,NRB,NCB,KA,KB)                           010661
C                                                    DEBUG #  56      010662
      IMPLICIT REAL*8(A-H,O-Z)                                        010663
      DIMENSION A(KA,1), B(KB,1), Z(KB,1)                             010664
      COMMON /ZBABKV/ W(500)                                        48010665
C                                                                     010666
C  SPECIAL TRIPLE MATRIX PRODUCT.  B*A*B(TRANSPOSE) = Z.              010667
C  A MUST BE SYMMETRIC TO GET CORRECT ANSWER.                         010668
C  Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF.   010669
C  THE MAXIMUM SIZE IS                                                010670
C      NCB = 500                                                      010671
C  DEVELOPED BY CARL BOSLEY.  JANUARY 1969.                           010672
C  LAST REVISION BY RE WOHLEN.  JULY 1972.                            010673
C                                                                     010674
C      SUBROUTINE ARGUMENTS                                           010675
C  A   = INPUT  INNER MATRIX. SIZE(NCB,NCB).                          010676
C  B   = INPUT  OUTER MATRIX. SIZE(NRB,NCB).                          010677
C  Z   = OUTPUT RESULT MATRIX. SIZE(NRB,NRB).                         010678
C  NRB = INPUT  NUMBER OF ROWS OF MATRIX B. SIZE OF MATRIX Z.         010679
C  NCB = INPUT  NUMBER OF COLS OF MATRIX B. SIZE OF MATRIX A. MAX=500.    010680
C  KA  = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.                010681
C  KB  = INPUT  ROW DIMENSION OF B,Z IN CALLING PROGRAM.              010682
C                                                                     010683
C                                                                     010684
      DO 40 J=1,NRB                                                   010685
      DO 20 L=1,NCB
      S = 0.0
      DO 10 K=1,NCB
   10 S = S + A(L,K)*B(J,K)
   20 W(L) = S
      DO 40 I=1,J
      S = 0.0
      DO 30 L=1,NCB
   30 S = S + B(I,L)*W(L)
      Z(I,J) = S
   40 Z(J,I) = S
      RETURN
      END
```

```
      SUBROUTINE MULTB (A,BZ,NRA,NRB,NCB,KA,KBZ)
C                                                           DEBUG #  57
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(KA,1),BZ(KBZ,1)
      COMMON /BLANKV/ W(500)
C
C MATRIX MULTIPLICATION.  A * B = Z.
C  USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.
C  BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C  LARGER OF B OR Z.
C  THE MAXIMUM SIZE IS
C       NRB = 500.
C
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----
C
C  A         = INPUT  MATRIX. SIZE (NRA,NRB).
C  BZ       = INPUT  MATRIX.  SIZE (NRB,NCB).
C           = OUTPUT RESULT MATRIX.  SIZE (NRA,NCB).
C  NRA      = INPUT  NUMBER OF ROWS OF MATRICES A,Z.
C  NRB      = INPUT  NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A.  MAX =
C  NCB      = INPUT  NUMBER OF COLS OF MATRICES B,Z.
C  KA       = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C  KBZ      = INPUT  ROW DIMENSION OF BZ IN CALLING PROGRAM.
C
      DO 40 J=1,NCB
      DO 20 K=1,NRB
   20 W(K) = BZ(K,J)
      DO 40 I=1,NRA
      S = 0.0
      DO 30 K=1,NRB
   30 S = S + A(I,K)*W(K)
   40 BZ(I,J) = S
C
      RETURN
      END
      SUBROUTINE REVADD (ALPHA,A,IVEC,JVEC,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)
C
C                                                     DEBUG #  98
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(KRA,1), IVEC(1), JVEC(1), Z(KRZ,1)
C
C  REARRANGE AND ADD ROWS AND COLUMNS OF ALPHA * MATRIX A INTO
C  MATRIX Z.
C  BE SURE MATRIX Z IS DEFINED BEFORE CALLING THIS SUBROUTINE. FOR
C  EXAMPLE, CALL ZERO TO CLEAR MATRIX Z.
C
C      SUBROUTINE ARGUMENTS
C  ALPHA = INPUT  SCALAR THAT MULTIPLIES MATRIX A.
C  A     = INPUT  MATRIX TO BE ARRANGED AND ADDED. SIZE(NRA,NCA).
C  IVEC  = INPUT  VECTOR. SIZE(NRA).
C                 IVEC(I)=ROW POSITION OF A(ROW I) IN Z.
C                 IF IVEC(I) IS PLUS ,Z=Z(ROW IVEC(I))+ALPHA*A(ROW I).
C                 IF IVEC(I) IS MINUS,Z=Z(ROW IVEC(I))-ALPHA*A(ROW I).
C                 IF IVEC(I) IS ZERO , A(ROW I) IS OMITTED IN Z.
C  JVEC  = INPUT  VECTOR. SIZE(NCA).
C                 JVEC(J)=COL POSITION OF A(COL J) IN Z.
C                 IF JVEC(J) IS PLUS ,Z=Z(COL JVEC(J))+ALPHA*A(COL J).
C                 IF JVEC(J) IS MINUS,Z=Z(COL JVEC(J))-ALPHA*A(COL J).
C                 IF JVEC(J) IS ZERO , A(COL J) IS OMITTED IN Z.
C  Z     = INPUT/OUTPUT MATRIX TO WHICH ALPHA*A IS ADDED. SIZE(NRZ,NCZ).
C  NRA   = INPUT  NUMBER OF ROWS IN MATRIX A.
C  NCA   = INPUT  NUMBER OF COLS IN MATRIX A.
C  NRZ   = INPUT  NUMBER OF ROWS IN MATRIX Z.
C  NCZ   = INPUT  NUMBER OF COLS IN MATRIX Z.
C  KRA   = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C  KRZ   = INPUT  ROW DIMENSION OF Z IN CALLING PROGRAM.
C
      DO 30 IA=1,NRA
      IZ = IABS(IVEC(IA))
      IF (IZ .EQ. 0)   GO TO 30
      DO 25 JA=1,NCA
      JZ = IABS(JVEC(JA))
      IF (JZ .EQ. 0)   GO TO 25
      SIGN = +1.D0
      IF (IVEC(IA).LT.0 .AND. JVEC(JA).GT.0  .OR.
     *    IVEC(IA).GT.0 .AND. JVEC(JA).LT.0) SIGN=-1.D0
      Z(IZ,JZ) = Z(IZ,JZ) + SIGN*ALPHA*A(IA,JA)
   25 CONTINUE
   30 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE [illegible]
C                                                     DEBUG #  99
      IMPLICIT REAL*8(A-H,O-Z)
C
C  SUBROUTINE SOLVES FOR THE LINEARIZED TIME RESPONSE.
C  [illegible] USING A [illegible] RUNGE-KUTTA STARTER AND AN ADAMS CORRECTOR
C  [illegible] FORMULA FOR THE [illegible] [illegible].
```

CONTENTS

Page

PREFACE . . . . iii 1/A5

I. PROGRAM SYSTEM OVERVIEW 1 1/A7

- A. Introduction 2 1/A8
- B. Simulation Overview and Nomenclature 3 1/A9
- C. General Use Information 8 1/A14

II. PROGRAM SEGMENTATION 9 1/B1

III. DELINEATION OF USER-SUPPLIED MODULES 10 1/B2

- A. Logic Flow 11 1/B3
- B. User-Supplied Module Specifics 12 1/B4
- C. Selected Common-Block Information 17 1/B9

IV. PROGRAM INPUTS 19 1/B11

- A. Subroutines READ and READIM 19 1/B11
- B. Input Data Stream 20 1/B12

V. PROGRAM OUTPUTS 20 1/B12

VI. AUXILIARY PROGRAMS 33 1/C11

- A. REDIM–The Redimension Program 33 1/C11
- B. Support Programs for the DISCOS Flexible-Body Capability 42 1/D6

APPENDIX A–INPUT/OUTPUT SUBROUTINE EXPLANATIONS A-1 1/F12

APPENDIX B–SUPPORT PROGRAMS FOR THE DISCOS FLEXIBLE-BODY CAPABILITY B-1 2/A5

```
C                                                                       010788
C        FOR OUTLINE OF PROCEDURES USED HERE IN SEE VOL 1,PAGE 85       010789
C              ' LINEAR TIME DOMAIN RESPONSE'                           010790
C                                                                       010791
      COMMON /KDSIZE/                                                   010792
     1                  KR, KRT, KRX, KV1, KV2, KVX                     010793
      COMMON /LDSIZE/                                                   010794
     2                  NX, NY, NDELTA, NXSS, NB, NDD, NY2, ND2         010795
      COMMON /TAPENO/                                                   010796
     4                  NOUT1, NOUT2, NOUT3                             010797
      COMMON /MISCNO/                                                   010798
     5                  NOPRNT, NOPLOT                                  010799
      COMMON /  LV1 /                                                   010800
     C                  V1  (100), V2  (100), V3  (100)               42610801
      COMMON /VECTOR/                                                   010802
     E                  Y   (250), YD  (250)                          43010803
      COMMON /LWORK1/                                                   010804
     F                  W1(100,100), W2(100,100)                      43210805
      COMMON /TIMESS/                                                   010806
     G                  ST, DT, T, ET, TMST                             010807
C                                                                       010808
C   ASSUMES THAT W1 = A* ON ENTRY.                                      010809
C   UNIT NOUT3 WILL BE WRITTEN FOR PLOTTING                             010810
C                                                                       010811
      DIMENSION PRK(4), YDS(250,3), YS(250)                           47010812
C                                                                       010813
C   STORE Y* IN YS, THEN ZERO Y.                                        010814
      DO 20 I=1,NX                                                      010815
      YS(I) = Y(I)                                                      010816
   20 Y(I) = 0.D0                                                       010817
      PRK(1) = 0.5D0                                                    010818
      PRK(2) = 1.D0 - DSQRT(0.5D0)                                      010819
      PRK(3) = 1.D0 + DSQRT(0.5D0)                                      010820
      PRK(4) = 0.5D0                                                    010821
C                                                                       010822
      NT = 0                                                            010823
      T = ST                                                            010824
      TMST = 0.D0                                                       010825
      IPRNT = 0                                                         010826
      IPLOT = 0                                                         010827
C                                                                       010828
      REWIND NOUT3                                                      010829
C                                                                       010830
      WRITE (NOUT3) ((W1(I,J),I=1,KR),J=1,KR)                           010831
      REWIND NOUT3                                                      010832
C                                                                       010833
      CON = 3.D0 * DT/8.D0                                              010834
C                                                                       010835
      DO 30 I=1,NX                                                      010836
      DO 30 J=1,NX                                                      010837
      W1(I,J) = -CON*W1(I,J)                                            010838
      IF (J .EQ. I) W1(I,J) = 1.+W1(I,J)                                010839
   30 CONTINUE                                                          010840
C                                                                       010841
      CALL GAUSSI (W1,W2,NX,KR)                                         010842
      READ (NOUT3) ((W1(I,J),I=1,KR),J=1,KR)                            010843
      REWIND NOUT3                                                      010844
C                                                                       010845
      CALL ZERO (V1,1,NX,1)                                             010846
      CALL LTORQL (V2)                                                  010847
      CALL YDOTL (W1,V2,Y,YD,NX,KK)
C
      DO 10 I=1,NX
   10 YDS(I,1) = YD(I)
C  USE THE R-K STARTER.
      CALL LPRNT
      CALL LPLTWR
C
  100 CONTINUE
      DO 120 J=1,4
      JIL = J
      DO 110 I=1,NX
      Z = YD(I) * DT
      GO TO (103,101,101,105),JIL
  101 R = PRK(JIL) * (Z-V1(I))
      GO TO 107
  103 R = PRK(JIL) * Z - V1(I)
      GO TO 107
  105 R = (Z-2.D0 * V1(I)) / 6.D0
  107 Y(I) = Y(I) + R
  110 V1(I) = V1(I) + 3.D0 * R - PRK(JIL) * Z
      IF (JIL .EQ. 1  .OR. JIL .EQ. 3) T = T+ DT/2.D0
      CALL LTCRGL (V2)
  120 CALL YDOTL (W1,V2,Y,YD,NX,KK)
C
C
      NT = NT + 1
      ANT = NT
      TMST = ANT * DT
      T = ST + TMST
C
      IPRNT = IPRNT + 1
      IF (IPRNT .NE. NOPRNT) GO TO 130
      CALL LPRNT
      IPRNT = 0
  130 CONTINUE
      IPLOT = IPLOT + 1
      IF (IPLOT .NE. NOPLOT) GO TO 140
      CALL LPLTWR
      IPLOT = 0
  140 CONTINUE
      DO 150 I=1,NX
  150 YDS(I,NT+1) = YD(I)
      IF (T .LE. ET  .AND.  NT .LT. 2) GO TO 100
C
C  THE ADAMS CORRECTOR FORMULA
C
      C0 = DT / 24.D0
      C1 = C0 * 9.D0
      C2 = C0 * 19.D0
      C3 = -C0 * 5.D0
      C4 = C0
C
C
C     ESTABLISH Y AT STEP NT
C
  200 CALL LTCRGL (V1)
C
C  V1 IS EXTERNAL FORCING FUNCTION FOR THE LINEAR SYSTEM.
C
```

```
      DO 210 I=1,NX
  210 Y(I) = Y(I) +C1*V1(I) + C2*YDS(I,3) + C3*YDS(I,2) + C4*YDS(I,1)
      CALL MULT3 (W2,Y,NX,NX,1,KR,KR)
C
C  RESET YDS FOR NEXT STEP.
C
      DO 220 I=1,NX
      YDS(I,1) = YDS(I,2)
  220 YDS(I,2) = YDS(I,3)
C
C  COMPUTE YD AT STEP NT.
      CALL YDCTL (W1,V1,Y,YD,NX,KR)
C
      DO 225 I=1,NX
  225 YDS(I,3) = YD(I)
C
      NT = NT + 1
      ANT = NT
      TMST = ANT * DT
      T = ST + TMST
C
      IPRNT = IPRNT + 1
      IF (IPRNT .NE. NOPRNT) GO TO 230
      CALL LPRNT
      IPRNT = 0
  230 CONTINUE
      IPLOT = IPLOT + 1
      IF (IPLOT .NE. NOPLOT) GO TO 240
      CALL LPLTWR
      IPLOT = 0
  240 CONTINUE
C
      IF (T .LT. ET) GO TO 200
C
C
C
      RETURN
      END
```

```
      SUBROUTINE LPLTWR
C                                                          DEBUG # 100
      IMPLICIT REAL*8(A-H,O-Z)
C
C  SUBROUTINE WRITES TAPE NUT3 FOR PLOTTING.
C
      COMMON /LDSIZE/
     2                NX, NY, NDLTA, NXSS, NB, NJJ, NY2, NDZ
      COMMON /TAPENO/
     4                NUT1, NUT2, NUT3
      COMMON /VECTOR/
     8                Y   (250), YD  (250)
      COMMON /TIMESS/
     6                ST, DT, T, ET, TMST
      COMMON /PLTDTA/
     1                 NRPLOT,NCPLOT
```

```
      DATA IIST / 0 /                                                   010962
      IF (IIST .NE. 0) GO TO 5                                          010963
      IIST = 1                                                          010964
      REWIND NOT3                                                       010965
      NRPLOT = 0                                                        010966
      NCPLOT = 2*NX + 1                                                 010967
    5 NRPLOT = NRPLOT + 1                                               010968
      WRITE (NOT3) T,(YD(I),I=1,NX),(Y(I),I=1,NX)                       010969
C                                                                       010970
      RETURN                                                            010971
      END                                                               010972




      SUBROUTINE LPRNT                                                  010973
C                                                      DEBUG # 101      010974
      IMPLICIT REAL*8(A-H,O-Z)                                          010975
C                                                                       010976
C   SUBROUTINE PRINTS OUT RESULTS OF LINEARIZED TIME RESPONSE.          010977
C                                                                       010978
      COMMON /VECTOR/                                                   010979
     E                 Y    (250), YD   (250)                         43010980
      COMMON /LDSIZE/ NX,NY,NDELTA,NXSS,NB,NJU,NY2,NU2                  010981
      COMMON /TIMESS/                                                   010982
     G                 ST, DT, T, ET, TMST                              010983
      DATA NOT/ 6 /                                                     010984
      DATA IIST/ 0 /                                                    010985
C                                                                       010986
      IF (IIST .NE. 0) GO TO 5                                          010987
      IIST = 1                                                          010988
C                                                                       010989
C   PRINT OUT DATA AT START.                                            010990
C                                                                       010991
      CALL PAGEHD                                                       010992
      WRITE (NOT,11)                                                    010993
   11 FORMAT (////30X,24HLINEARIZED TIME RESPONSE /                     010994
     *              30X,24H GENERAL INFORMATION      //)                010995
      WRITE (NOT,12) ST,DT,ET                                           010996
   12 FORMAT (//30X,26HINTEGRATION PARAMETERS ARE  //                   010997
     *             30X,13HSTART TIME = D12.5,/,                         010998
     *             30X,13HDELTA TIME = D12.5,/,                         010999
     *             30X,13HEND   TIME = D12.5)                           011000
C                                                                       011001
      LY = 1                                                            011002
      LX = LY+NY2                                                       011003
      LU=LX+NXSS                                                        011004
      LS=LU+NU2                                                         011005
C                                                                       011006
    5 CONTINUE                                                          011007
      CALL PAGEHD                                                       011008
      WRITE (NOT,101) T                                                 011009
  101 FORMAT  (/5X,18HSIMULATION TIME = D12.5)                          011010
C                                                                       011011
      WRITE (NOT,102) NY2,NXSS,NU2,NB                                   011012
  102 FORMAT  (/30X,30HNUMBER OF PLANT VARIABLES    = I5,               011013
     *           /30X,30HNUMBER OF SENSOR SIGNALS   =  I5,              011014
     *           /30X,30HNUMBER OF CONTROL VARIABLES = I5,              011015
     *          /30X,30HNUMBER OF CONTROL TORQUES    = I5)                011016
C                                                                         011017
      WRITE (NCT,103)                                                     011018
  103 FORMAT (//20X,29HSTATE VECTOR TIME DERIVATIVES /)                   011019
      CALL WRITES (YD,1,NX,1)                                             011020
C                                                                         011021
      WRITE (NCT,104)                                                     011022
  104 FORMAT  (/20X,12HSTATE VECTOR /)                                    011023
      CALL WRITES (Y ,1,NX,1)                                             011024
C                                                                         011025
      WRITE (NCT,105)                                                     011026
  105 FORMAT (/20X,20HPLANT VARIABLE RATES )                              011027
      CALL WRITES (YD,1,NY2,1)                                            011028
C                                                                         011029
      WRITE (NCT,106)                                                     011030
  106 FORMAT (//20X,19HSENSOR SIGNAL RATES)                               011031
      CALL WRITES (YD(LX),1,NXSS,1)                                       011032
C                                                                         011033
      WRITE (NCT,107)                                                     011034
  107 FORMAT (//20X,22HCONTROL VARIABLE RATES)                            011035
      CALL WRITES (YD(LD),1,ND2,1)                                        011036
C                                                                         011037
      WRITE (NCT,108)                                                     011038
  108 FORMAT (//20X,19HTORQUE SYSTEM RATES)                               011039
      CALL WRITES (YD(LB),1,NB,1)                                         011040
C                                                                         011041
      WRITE (NCT,109)                                                     011042
  109 FORMAT (///20X,20HPLANT VARIABLE STATE)                             011043
      CALL WRITES (Y,1,NY2,1)                                             011044
C                                                                         011045
      WRITE (NCT,110)                                                     011046
  110 FORMAT (//20X,19HSENSOR SIGNAL STATE)                               011047
      CALL WRITES (Y(LX),1,NXSS,1)                                        011048
C                                                                         011049
      WRITE (NCT,111)                                                     011050
  111 FORMAT (//20X,22HCONTROL VARIABLE STATE)                            011051
      CALL WRITES (Y(LD),1,ND2,1)                                         011052
C                                                                         011053
      WRITE (NCT,112)                                                     011054
  112 FORMAT (//20X,19HTORQUE SYSTEM STATE)                               011055
      CALL WRITES (Y(LB),1,NB,1)                                          011056
C                                                                         011057
      RETURN                                                              011058
      END                                                                 011059
```

```
      SUBROUTINE YDOTL (A,B,Y,YD,NY,KA)                                   011060
C                                                       DEBUG # 102       011061
      IMPLICIT REAL*8(A-H,O-Z)                                            011062
C                                                                         011063
C   SUBROUTINE FORMS THE LINEARIZED YDOT VECTOR.                          011064
C                                                                         011065
C               -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                011066
C                                                                         011067
C   A       = INPUT  LINEARIZED COEFFICIENTS.  SIZE NA BY NY.             011068
C                     USED IN EXPRESSION YD = A * Y + B                   011069
```

```
C  B       = INPUT EXTERNAL FORCING TORQUES.  USER SUPPLIED VIA                     011070
C                   SUBROUTINE ETORQE.                                              011071
C  Y       = INPUT VECTOR OF STATE VARIABLES.                                       011072
C  YD      = OUTPUT VECTOR OF STATE VECTOR TIME DERIVATIVES.                        011073
C  NY      = SIZE OF STATE VECTOR TO BE INTEGRATED.                                 011074
C                                                                                   011075
       DIMENSION A(KA,1),B(1),Y(1),YD(1)                                            011076
C                                                                                   011077
      DO 10 I=1,NY                                                                  011078
      YD(I) = B(I)                                                                  011079
      DO 10 J=1,NY                                                                  011080
      YD(I) = YD(I) + A(I,J) * Y(J)                                                 011081
   10 CONTINUE                                                                      011082
      RETURN                                                                        011083
      END                                                                           011084
```

```
      SUBROUTINE ETORQE (VTORQ)                                                     011085
C                                                          DEBUG # 103              011086
      IMPLICIT REAL*8(A-H,O-Z)                                                      011087
C****************   USER SUPPLIED SUBROUTINE   *******                             011088
C                                                                                   011089
C        VTORQ ARRAY TO CONTAIN THE ELEMENTS B IN THE EQUATION                      011090
C                    Y(DOT) = A*Y + B                                               011091
C        THE LINEARIZED EQUATIONS OF MOTION                                         011092
C                                                                                   011093
      DIMENSION VTORQ(1)                                                            011094
      COMMON /KDSIZE/                                                               011095
     1                  KR, KRT, KRA, KV1, KV2, KVX                                 011096
      COMMON /VECTOR/                                                               011097
     E                  Y   (250), YD  (250)                                      43011098
      COMMON /TIMESS/                                                               011099
     C                  ST, DT, T, ET, TMST                                         011100
C                                                                                   011101
C                                                                                   011102
      CALL ZERO (VTORQ,1,KVX,1)                                                     011103
      RETURN                                                                        011104
      END                                                                           011105
```

```
      FUNCTION ADT (IC,T)                                                           011106
C                                                          DEBUG # 104              011107
C        USER DEFINED FUNCTION                                                      011108
C          USED TO DEFINE THE ALPHA DOT TERM IN EQUATION II-5 VOL 1                 011109
C        THE VELOCITY OF THE COORDINATE IC WHICH IS RHENOMICALLY                    011110
C        CONSTRAINED IS DEFINE HERE. SEE SUBROUTINE FINDU                           011111
C        SEE ADDT GET THIS TERM VIA INTEGRATION OF ADDT IN CONTRL                   011112
C              NUMERICAL PROBLEMS MAY ARISE IF THIS IS NOT DONE                     011113
      IMPLICIT REAL*8 (A-H,O-Z)                                                     011114
            COMMON /VECTOR/                                                         011115
     *      Y(250),YDT(250)                                                        2011116
C                                                                                   011117
```

```
      RETURN
C
      END
```

```
      FUNCTION ADDT (IC,T)
C                                                         DEBUG # 105
C        USER DEFINED FUNCTION
C        USED TO DEFINE ACCELERATION OF RHEONOMICALLY CONSTRAINED
C        COORDINATE IC, SEE SUBROUTINE YDOT
C        THE ALPHA DOUBLE DOT TERM IN EQUATION II-6
C        CAUTION INTEGRATE ADDT IN CONTRL TO GET ADT.
C          STEP CHANGES IN VELOCITY NUMERICALLY LEAD TO EXTRANEOUS
C          IMPULSES
      IMPLICIT REAL*8 (A-H,O-Z)
            COMMON /VECTOR/
     *      Y(250),YDT(250)
C
      RETURN
      END
```

```
      SUBROUTINE KHINGE (G)
C                                                         DEBUG # 106
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION G(1)
C
            COMMON /BHDSRJ/
     *      BH(6,16,11),BS(6,18,15),RJL(3,3, 6),DJL(3, 6)
            COMMON /CONPAR/
     *      CNTLTA(150)
            COMMON /MAXMUM/
     *      NLMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMOBJD,KMJ,KY,KQ
            COMMON /MOMENG/
     *      P(113),PMOM(30),HTOT(3),TOTL(3),ENGKE( 6),ENGPE( 6),
     *      TOTKE, TOTPE, TOTENG, AHTOT,ATOTL
            COMMON /SPECIF/
     *      DELTAH(6, 6),DELTAHD(6, 6),AMD(2, 5),RH(3,3,30),RS(3,3,30),
     *      DH(3,35),DS(3,30),IMD(3, 5),NNOW(6, 6),IFTSMW(15),
     *      NO,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 6),IRSFLX( 6),IHDATA(7, 6),
     *      LOCO(14),LENO(14),NQ,NBETA,NLAM,NEQ
C
      DIMENSION SK(3,6),DK(3,6)
      DIMENSION HNGT(6, 6)
C        ZERO OUT HNGT ARRAY AND POTENTIAL ENERGY TERM
      DO 11 L=1,NH
      DO 11 I=1,6
   11 HNGT(I,L) = 0.D0
      TOTPE = 0.D0
C
C
C************************ USER
```

```
C         TO SET UP HINGE TORQUES DUE TO SPRINGS AND DASHPOTS ACROSS
C         GIMBAL AXES
C         1) EQUIVALENCE ARRAYS WHICH HOLD CONSTANTS TO PROPER AREA
C            OF DATA BLOCK CNTDTA
C         2) SET HINGE TORQUE ABOUT EACH AXIS
C         3) COMPUTE POTENTIAL ENERGY
C         EXAMPLE
C     EQUIVALENCE (CNTDTA(12),SK(1)), (CNTDTA(30),DK(1))
C     DO 10 L=1,NH
C     DO 10 I=1,3
C     FNGT(I,L) = -(SK(I,L)*BETAH(I,L) + DK(I,L)*BETAHD(I,L))
C  10 TOTPE = TOTPE + 0.500*SK(I,L)*BETAH(I,L)**2
C
C         ADD HINGE TORQUE/FORCE  EFFECTS INTO BODY 1 G VECTOR
      LEQ = IRGFLX(1) + 6
      DO 15 I=1,6
      F = FNGT(I,1)
      DO 16 J=1,LEQ
   16 G(J) = G(J) + F*GH(I,J,1)
   15 CONTINUE
C
C         ADD HINGE TORQUE/FORCE  EFFECTS TO G VECTOR FOR EACH HINGE
      DO 20 L=2,NH
      NBBQ = ITOPOL(1,L)
      NBBP = ITOPOL(2,L)
      LQ = 2*L - 2
      LP = LQ + 1
      LOQ = LOCB(NBBQ) - 1
      LOP = LOCB(NBBP) - 1
      LEQ = IRGFLX(NBBQ) + 6
      LEP = IRGFLX(NBBP) + 6
      DO 20 I=1,6
      F = FNGT(I,L)
      DO 25 J=1,LEQ
      LOQJ = LOQ + J
   25 G(LOQJ) = G(LOQJ) + F*GH(I,J,LQ)
      DO 26 J=1,LEP
      LOPJ = LOP + J
   26 G(LOPJ) = G(LOPJ) + F*GH(I,J,LP)
   20 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CONTRL
C                                                    DEBUG # 107
      IMPLICIT REAL*8 (A-H,O-Z)
C         THIS ROUTINE CONTRL MUST BE USER DEFINED
C         IT IS USED TO ESTABLISH THE TIME DERIVATIVES OF THE CONTROL
C         SYSTEM STATE VARIABLES
C
            COMMON /GHCBRD/
     *      GH(6,13,11),BL(3,18,13),RDL(3,3, 6),DDL(3, 6)
            COMMON /CONPAR/
     *      CNTDTA(150)
```

```
      COMMON /LOSIZE/ NX,NY,NDLTA,NXSS,NBTQ,NJQ,NY2,ND2
            COMMON /SPECIF/
     *     BETAH(6, 6),BETAHD(6, 6),AMD(2, 5),RH(3,3,30),RS(3,3,30),
     *     DH(3,35),DS(3,30),IMD(3, 5),NMOW(6, 5),IFISMW(15),
     *     NB,NH,NSPT,NGFMD,NDELTA,ITOPOL(2, 6),IRGFLX( 6),IHDATA(7, 6),
     *     LOCC(14),LENG(14),NG,NBETA,NLAM,NEQ
            COMMON /TIMESS/
     *     STARTT,DELTAT,T,ENDT,IMST
            COMMON /VECTOR/
     *     Y(250),YDT(250)
CCCCCCC  THIS COMMON IS TRANSFER BETWEEN CONTRL AND SHAFTT ONLY ----
C           COMMON /WHEEL /
C    *     CLM(4)
C        IF DATA IS TO BE COMPUTED WHICH MUST BE TRANSFERED TO OTHER
C        USER DEFINED SUBROUTINES ADD ADDITIONAL SPECIAL PURPOSE
C        COMMON BLOCKS
C
C        FUNCTIONS REQUIRED BY SAMPLE PROBLEM
C     ALIM(U,V) = DMAX1(-V,DMIN1(U,V))
C
C        ADD DIMENSION STATEMENTS FOR CONTROL SYSTEM DESCRIPTION
C     DIMENSION  TQ(6),TQD(6),RHD(3),THADW(3)
C     DATA ICT4/0/, RHD / 0.D0, 0.D0, 0.D0 /
C
C
C        SET UP DIMENSION STATEMENTS AND DEFAULT VALUES FOR TRANSFER
C        FUNCTION INPUT DIRECTLY INTO CONTRL
C              NPLY = NUMBER OF POLYNOMIAL RATIO PAIRS (NUMERATOR AND
C                     DENOMIATOR) TO BE READ ON FIRST PASS THRU CONTRL
C              KRY = ROW DIMENSION OF CPLY (HIGHEST DENOMINATOR ORDER + 1
C              KCY = COLUMN DIMENSION OF CPLY (2 * NUMBER OF POLYNOMIAL
C                    TRANSFER FUNCTIONS TO BE READ)
C              CPLY(I,2*K-1) = DENOMINATOR COEFFICIENTS IN ASCENDING
C                              ORDER FOR INPUT POLY K, I=1,2,...,KPLY(K)
C              KPLY(K)-1 = DEGREE OF POLY TRANSFER FUNCTION DENOMINATOR
C                          ALSO NUMBER OF FIRST ORDER DIFFERENTIAL
C                          EQUATIONS SET UP IN TFPLY FOR K-TH TRANSFER
C                          FUNCTION
C              CPLY(I,2*K) = NUMERATOR COEFFICIENTS ASCENDING ORDER
C        SEE SUBROUTINE DEFS,'SUBROUTINE DYNS2D INPUT...DATA'
      DIMENSION CPLY(10,4), KPLY(2), CI(2)
      DATA NPLY, KRY, KCY/ 0, 10, 4/
      DATA IIST/ 0 /
C
CCCCCCCCCC
CCCCCCCCCC
CCC  THE FOLLOWING STATEMENTS MUST ALWAYS BE IN CONTRL..
C          FIRST PASS THROUGH READ IN ALL CONTROL POLY TRANSFER FUNCTIONS
      IF (IIST .NE. 0) GO TO 110
      IIST = 1
      IF (NPLY .EQ. 0) GO TO 106
      CALL ZERO (CPLY,KRY,KCY,KRY)
      DO 105 K=1,NPLY
         K2=2*K-1
  105 CALL READ (CPLY(1,K2),KPLY(K),N2,KRY,KCY)
      CALL WRITE (CPLY,KRY,KCY,4HCPLY,KRY)
  106 CONTINUE
      NDLTA = NDELTA
      LDFL = LOCC(2*NB+2) - 1
  110 CONTINUE
```

```
C           DEFAULT VALUES                                                   011280
C           INTERNAL DO LOOP LOGIC REQUIRES THAT                             011281
C              NACX = NRSS + NBTQ                                            011282
C           WHERE                                                            011283
C              NACX > 0                                                      011284
C           AND IT MUST BE REDEFINED IN (JA3)                                011285
      NRSS = 1                                                               011286
      NBTQ = 0                                                               011287
C           NRSS = NUMBER OF PLANT SENSOR SIGNALS                            011288
C           NBTQ = NUMBER OF CONTROL SYSTEM OUTPUTS                          011289
C           THESE MUST BE USER DEFINED IF NON-ZERO. FOR EXAMPLE              011290
C     NRSS = 3                                                               011291
C     NBTQ = 3                                                               011292
      IF (NDELTA .EQ. 0) RETURN                                              011293
CCCCCCCCCC     CCC                                                           011294
CCCC---NOTE---THIS SUBROUTINE MUST ESTABLISH  NDELTA, NRSS AND NBTQ          011295
CCCCCCCCCCC                                                                  011296
C                                                                            011297
CCCC  ESTABLISH THE DZDT(DELTAS)                                             011298
C                                                                            011299
CCCCCCCCCC                                                                   011300
CCCC---NOTE---THIS SECTION IS TYPICAL OF USE OF T-PLY.                       011301
CCCCCCCCCC                                                                   011302
C                                                                            011303
      IF (NPLY .EQ. 0) GO TO 116                                             011304
      L = LDEL+1                                                             011305
C           L = STARTING LOCATION FOR FIRST DELTA DOT EQUATION               011306
      DO 115 K=1,NPLY                                                        011307
         K2 = 2*K-1                                                          011308
C           CALL TFPLY TO SETUP DIFFERENTIAL EQUATIONS ASSOCIATED WITH POLY  011309
C           TRANSFER FUNCTIONS                                               011310
      CALL TFPLY (CPLY(1,K2),CPLY(1,K2+1),UI(K),K,KPLY(K),L)                 011311
      L = L+KPLY(K) - 1                                                      011312
  115 CONTINUE                                                               011313
C           KPLY(1) + KPLY(2) + ...+ KPLY(NPLY) - NPLY = THE NUMBER OF       011314
C           FIRST ORDER EQUATIONS DEFINED AND PUT IN YDT ARRAY               011315
  116 CONTINUE                                                               011316
C                                                                            011317
CCCCCCCCCC                                                                   011318
C           EXAMPLE CONTROL SYSTEM FOR ATS-F                                 011319
C     ICT4 = ICT4 + 1                                                        011320
C     IA = (ICT4-1)/4                                                        011321
C     IAA = (ICT4-2)/4                                                       011322
C     IFLAG = IA - IAA                                                       011323
C     DO 6 I=1,3                                                             011324
C   6 THADV(I) = Y(6+I)                                                      011325
C     DO 5 I=1,6                                                             011326
C   5 TQ(I) = Y(LDEL+I)                                                      011327
C                                                                            011328
C            WHEEL 1 (ROLL INERTIA WHEEL CONTROL TORQUE)                     011329
C            DEFINE DIFFERENTIAL EQUATIONS FOR ROLL CONTROL LOOP             011330
C                                                                            011331
C     U1 = 57.2958D0*ROL(3,2,1)/ROL(3,3,1)                                   011332
C     U5 = ALIM(TQ(5),29.D0)                                                 011333
C     U2 = 2.17D3*U1 - U5                                                    011334
C     U3 = ALIM(1.1D0*U2,1.17D0)                                             011335
C     TQD(5) = (1.D0/88.D0)*(-TQ(5) + (571.1D0)*U3)                          011336
C     U6 = ALIM(5*U3,1.56D0)                                                 011337
C     U8 = ALIM(TQ(6),1.9D0)                                                 011338
C     IF (IFLAG .EQ. 0) GO TO 32                                             011339
```

```
C      U0 = DABS(U8)
C      IF (U0.GT.1.D0) GO TO 30
C      IF (U0.LT.0.5D0) GO TO 31
C      U9 = RHO(1)
C      GO TO 10
C   30 U9 = U8/U0
C      GO TO 10
C   31 U9 = 0.D0
C      GO TO 10
C   32 U9 = RHO(1)
C      GO TO 33
C   10 RHO(1) = U9
C   33 CONTINUE
C      TQD(6) = (-TQ(6) + 2.5D0*(U8-U9))/.5D0
C
C   1500 RPM = 157.0795 RAD/SEC
C    1 INCH*OZ = .03125 FT*LBS
C
C      IF (DABS(THADW(1)).GT. 157.0795D0) U9 = 0.D0
C      CLM(1) = .03125D0*U9 - 5.D-05*THADW(1)
C
C           WHEEL 2 (PITCH INERTIA WHEEL CONTROL TORQUE)
C           DEFINE DIFFERENTIAL EQUATIONS IN PITCH CONTROL LOOP
C
C      U1 = -57.2958D0*ROL(3,1,1)/ROL(3,3,1)
C      U5 = ALIM(TQ(1),16.4D0)
C      U2 = 2.17D0*U1 - U5
C      U3 = ALIM(.32D0*U2,1.17D0)
C      TQD(1) = (-TQ(1) + U3*(77.82D0))/50.D0
C      U6 = ALIM(5*U3,1.66D0)
C      U8 = ALIM(TQ(2),1.5D0)
C      IF (IFLAG.EQ.2) GO TO 14
C      U0 = DABS(U8)
C      IF (U0.GT. 1.D0) GO TO 15
C      IF (U0.LT.0.5D0) GO TO 16
C      U9 = RHO(2)
C      GO TO 12
C   15 U9 = U8/U0
C      GO TO 12
C   16 U9 = 0.D0
C      GO TO 12
C   14 U9 = RHO(2)
C      GO TO 13
C   12 RHO(2) = U9
C   13 CONTINUE
C      TQD(2) = (-TQ(2) + 2.5D0*(U8 - U9))/.5D0
C      IF (DABS(THADW(2)).GE. 157.0795D0) U9 = 0
C      CLM(2) = .03125D0*U9 - 5.D-05*THADW(2)
C
C           WHEEL 3 (YAW INERTIA WHEEL CONTROL TORQUE)
C           DEFINE DIFFERENTIAL EQUATIONS FOR YAW CONTROL LOOP
C
C      U1 = 57.2958D0*ROL(2,1,1)/ROL(2,2,1)
C      U2 = ALIM(U1,2.D0)
C      U6 = ALIM(TQ(3),29.D0)
C      U3 = 2.17D0*U2 - U6
C      U4 = ALIM(1.47D0*U3,1.17D0)
C      TQD(3) = (1.D0/88.D0)*(-TQ(3) + (971.47D0)*U4)
C      U7 = ALIM(5*U4,1.66D0)
C      U9 = ALIM(TQ(4),1.9D0)
```

```
C      IF (IFLAG.EQ.0) GO TO 20                                          011400
C      UU = DABS(U9)                                                     011401
C      IF (UU.GT.1.D3) GO TO 21                                          011402
C      IF (UU.LT. 0.5D0) GO TO 22                                        011403
C      U10 = RHO(3)                                                      011404
C      GO TO 18                                                          011405
C   21 U10 = U9/UU                                                       011406
C      GO TO 18                                                          011407
C   22 U10 = 0.D0                                                        011408
C      GO TO 18                                                          011409
C   20 U10 = RHO(3)                                                      011410
C      GO TO 24                                                          011411
C   18 RHO(3) = U10                                                      011412
C   24 CONTINUE                                                          011413
C      TQD(4) = (-TQ(4) + 2.5D0*(U7 - U10))/.5D0                         011414
C      IF (DABS(THADW(3)) .GT. 157.0795D0) U10 = 0.D0                    011415
C      CLM(3) = .03125D0*U10 - 5.D-05*THADW(3)                           011416
C                                                                        011417
C         PUT ALL CONTROL SYSTEM EQUATION IN YDT ARRAY                   011418
C      DO 34 I=1,6                                                       011419
C   34 YDT(LDEL+I) = TQD(I)                                              011420
C      YDT(LDEL+7) = Y(16)                                               011421
C      SK4 = CNTDTA(NDELTA+1)                                            011422
C      DK4 = CNTDTA(NDELTA+2)                                            011423
C      CLM(4) =-(SK4*Y(LDEL+7) + DK4*YDT(LDEL+7))                        011424
C                                                                        011425
       RETURN                                                            011426
       END                                                               011427




       SUBROUTINE EXTOR  (TEX,ISPN,NTEX)                                 011428
C                                                      DEBUG # 108       011429
       IMPLICIT REAL*8 (A-H,O-Z)                                         011430
       DIMENSION TEX(6,1), ISPN(1)                                       011431
C                                                                        011432
               COMMON /MAXNUM/                                           011433
     *     NJMAX,NHMAX,NSPMAX,NMWMAX,NMWDDD,NMDDDD,KMD,KY,KU             011434
               COMMON /SPECIF/                                           011435
     *     DETAM(6, 6),DETAMD(6, 6),ARC(2, 5),RR(3,3,30),RS(3,3,30),   1611436
     *     SM(3,30),SS(3,30),IMG(3, 3),NMJW(6, 6),IPTSMW(15),          1711437
     *     NJ,NM,NSPT,NSPMC,NDELTA,ITOPCL(2, 6),ITOPLX( 6),IHDATA(7, 6), 1811438
     *     LDEL(14),LEND(14),NU,NDLTA,NLAM,NEQ                         1911439
               COMMON /VECTOR/                                           011440
     *     Y(250),YDT(250)                                              2011441
C                                                                        011442
       DATA IIST / 0 /                                                   011443
C                                                                        011444
CCC ESTABLISH THE EXTERNAL FORCE/TORQUE (6-LONG VECTOR) AND NUMBER       011445
CCC  THE CORRESPONDING SENSOR POINTS. ALSO ESTABLISH THE NUMBER OF       011446
CCC  SIX-LONG VECTORS (NTEX).                                            011447
C          SEE VOL II,PAGE 12                                            011448
C        ISPN = INTEGER ARRAY TO DEFINE WHICH SENSOR POINTS ARE TO BE    011449
C               USED FOR FORCE/TORQUE INPUTS                             011450
C                                                                        011451
       IF (IIST .EQ. 1) GO TO 5                                          011452
       IIST = 1                                                          011453
```

```
      DO 10 I=1,6                                                       011454
      DO 10 J=1,NSPMAX                                                  011455
   10 TEX(I,J) = 0.D 0                                                  011456
C                                                                       011457
    5 NTEX = 0                                                          011458
C                                                                       011459
      RETURN                                                            011460
      END                                                               011461




      SUBROUTINE SHAFTT (TSHFT)                                         011462
C                                                     DEBUG # 109       011463
      IMPLICIT REAL*8 (A-H,O-Z)                                         011464
      DIMENSION TSHFT(1)                                                011465
C                                                                       011466
             COMMON /MAXMUM/                                            011467
     *     NJMAX,NHMAX,NSPMAX,NM*MAX,NM*DOD,NMD33D,K4U,KY,KU            011468
             COMMON /SPECIF/                                            011469
     *     BETAH(6, 6),BETAHD(6, 6),AMO(2, 5),RH(3,3,33),RS(3,3,30),   1611470
     *     CH(3,35),DS(3,33),IMO(3, 5),NMC#(6, 5),IFTSM#(15),          1711471
     *     NJ,NH,NSPT,NOFMD,NDELTA,ITOPOL(2, 6),IRSFLX( 6),IFDATA(7, 6), 1811472
     *     LOCO(14),LENO(14),NU,NBETA,NLAW,NEQ                         1911473
             COMMON /VECTOR/                                            011474
     *     Y(250),YDT(250)                                             2011475
C                                                                       011476
      DATA IFST / 0 /                                                   011477
C                                                                       011478
      IF (IFST .EQ. 1) GO TO 20                                         011479
      IFST = 1                                                          011480
      DO 5 I=1,NM*MAX                                                   011481
    5 TSHFT(I) = 0.D 0                                                  011482
   20 CONTINUE                                                          011483
C                                                                       011484
C         INTERFACE WITH SUBROUTINE CONTRL, WHEEL TORQUES COMPUTED      011485
C         IN CONTRL STORED IN CLM ARRAY, DISCOS NEEDS THEM IN           011486
C         THE TSHFT ARRAY                                               011487
C     EXAMPLE:                                                          011488
C         1) SETUP COMMON BLOCK TO TRANSFER DATA                        011489
C         2) EQUATE TORQUE ARRAY CLM TO TSHFT TORQUE ARRAY              011490
C                                                                       011491
CCCCCCC  THIS COMMON IS TRANSFER BETWEEN CONTRL AND SHAFTT ONLY ----    011492
C              COMMON /WHEEL /                                          011493
C    *     CLM(4)                                                       011494
C  10 DO 15 I=1,4                                                       011495
C  15 TSHFT(I) = CLM(I)                                                 011496
C                                                                       011497
      RETURN                                                            011498
      END                                                               011499




      SUBROUTINE EQADD                                                  011500
C                                                     DEBUG # 110       011501
```

```
      IMPLICIT REAL*8 (A-H,O-Z)                                          011502
C                                                                        011503
C        SEE VOL II, PAGE 15                                             011504
C        AUXILARY EQUATIONS WHICH DEFINE THE PHYSICALLY REALIZABLE       011505
C        SENSOR SIGNALS AND CONTROL TORQUES ARE TO BE PLACED IN THE      011506
C        YDT ARRAY AFTER THE LAST EQUATION, THAT IS YDT(NEQ)             011507
C        THE ORDER ASSUMED IS SENSOR SIGNAL, TORQUE SIGNAL               011508
            COMMON /DDGSRO/                                              011509
     *     SR(2,18,11),HS(6,15,15),ROL(3,3, 6),DDL(3, 6)                 211510
            COMMON /DNAUX /                                              011511
     *    NAUX                                                           011512
      COMMON /EQSIZE/ NX,NY,NDELTA,NXSS,NHTQ,NJQ,NY2,NJ2                 011513
            COMMON /MAXNUM/                                              011514
     *    NBMAX,NHMAX,NGMMAX,NMAMAX,NYAHLD,NMODSJ,NAU,NY,NU              011515
            COMMON /SPECIF/                                              011516
     *    BETAH(6, 6),BETAHD(6, 6),AMC(2, 6),RH(3,3,30),RS(3,3,30),     1611517
     *    GH(3,35),ZS(3,30),IHG(3, 3),NNGA(6, 6),I-TSAM(15),            1711518
     *    NS,NH,NSPT,NGMMO,NDBETA,ITOPLE(2, 6),IRGFLX( 6),IHDATA(7, 6), 1811519
     *    LOCC(14),LEND(14),NC,NBETA,NLAM,NEQ                           1911520
            COMMON /VECTOR/                                              011521
     *    Y(250),YDT(250)                                               2011522
C                                                                        011523
C        THIS SUBROUTINE MUST DEFINE NAUX                                011524
C        NAUX = NUMBER OF AUXILARY EQUATIONS, DEFAULT = 1                011525
C              FOR THE LINEARIZATION PORTION OF DISCOS AT LEAST ONE      011526
C              AUXILIARY EQUATION REQUIRED, IT MUST BE A FUNCTION        011527
C              OF THE SYSTEM STATE VARIABLES                             011528
C        THESE MUST BE COMPATABLE WITH THAT DEFINED IN CONTRL            011529
C        IN CONTRL THE DEFAULT VALUES ARE:                               011530
C            NXSS = 1                                                    011531
C            NHTQ = 0                                                    011532
      NAUX = NXSS + NHTQ                                                 011533
C        DEFAULT EQUATION FIRST UNCONSTRAINED BETA ANGLE                 011534
      YDT(NEQ+1) = Y(LOCC(2*NB+1))                                       011535
C                                                                        011536
C        SET OF AUXILARY EQUATIONS FOR OBTAINING TRANSFER FUNCTIONS      011537
C        BETWEEN PHYSICALLY REALIZABLE SENSOR SIGNALS AND APPLIED        011538
C        TORQUES                                                         011539
C     EXAMPLE:                                                           011540
C     NAUX = 6                                                           011541
C     LDEL = LOCC(2*NB+2) - 1                                            011542
C     ACCN = 57.295780                                                   011543
C     YDT(NEQ+1) =  ACCN*ROL(3,2,1)/ROL(3,3,1)                           011544
C     YDT(NEQ+2) = -ACCN*ROL(3,1,1)/ROL(3,3,1)                           011545
C     YDT(NEQ+3) =  ACCN*ROL(2,1,1)/ROL(2,2,1)                           011546
C     YDT(NEQ+4) = Y(LDEL+2)                                             011547
C     YDT(NEQ+5) = Y(LDEL+4)                                             011548
C     YDT(NEQ+6) = Y(LDEL+6)                                             011549
      RETURN                                                             011550
      END                                                                011551




      SUBROUTINE SMISC (N,LL,LC,V2)                                      011552
C                                                          DEBUG # 111   011553
      IMPLICIT REAL*8 (A-H,O-Z)                                          011554
      DIMENSION V2(1)                                                    011555
```

```
C                                                                            011556
              COMMON /MAXMUM/                                                011557
     *     NUMAX,NHMAX,NSPMAX,NMWMAX,NMWEGD,NMODDD,KMD,KY,KU                 011558
              COMMON /SPECIF/                                                011559
     *     BETAH(6, 6),BETAHD(6, 6),AMD(2, 5),RH(3,3,30),RS(3,3,30),       1611560
     *     DH(3,35),DS(3,30),IMD(3, 5),NMDW(6, 5),IFTSMW(15),              1711561
     *     NJ,NH,NSPT,NGFMD,NDELTA,ITCPUL(2, 6),IRGFLX( 6),IHDATA(7, 6),   1811562
     *     LOCC(14),LEND(14),NJ,NBETA,NLAM,NEQ                             1911563
              COMMON /VECTOR/                                                011564
     *     Y(250),YDT(250)                                                 2011565
C                                                                            011566
      DATA IIST / 0 /                                                        011567
C                                                                            011568
CCC   USER SUPPLIED SUBROUTINE TO CREAT MISC. CONTRIBUTIONS TO R.H.S.        011569
CCC   INCLUDING  THE THERMAL GRADIENT ENVIRONMENT.                           011570
C        IF THERMALLY INDUCED MOTION MODEL REQUIRED REPLACE 0.D0 BY          011571
C        THE THERMAL POSITION THE PARTICULAR MODE IN THE DO LOOP             011572
C        APPROPRIATE COMPUTATION OF THE THERMAL POSITION MUST BE             011573
C        CARRIED OUT, PREFERABLE IN CONTRL DATA TRANSFERED BY A              011574
C        SPECIAL PURPOSE COMMON BLOCK                                        011575
      IF (IIST .EQ. 1) GO TO 5                                               011576
      IIST = 1                                                               011577
C                                                                            011578
    5 CONTINUE                                                               011579
      LCM1 = LC - 1                                                          011580
      DO 10 I=1,LL                                                           011581
   10 V2(I) = Y(LCM1 + I) - 0.D 0                                            011582
C                                                                            011583
      RETURN                                                                 011584
      END                                                                    011585
```

```
      SUBROUTINE DEFI                                                        011586
C                                                       DEBUG # 112          011587
C                                                                            011588
C                                                                            011589
C                                                                            011590
C                                                                            011591
C                                                                            011592
C                           COMMON BLOCK / SUBROUTINE                        011593
C                               CROSS-REFERENCE                              011594
C                                    MATRIX                                  011595
C                                                                            011596
C         ABCDDGGHIIIJKLLLLLLLLLLLLLLLL MMMNNPPPPQSSTTVV                     011597
C         WHONRGGALNVIDACBDDIRRSSITTVVWWZAIOHUILRSPPJAIEI                    011598
C         LENAADGNITCLSMODESJAUTTIOI20RMXSMNMNTWTREMPMCN                     011599
C         ESPUTAADNGOFIBUABIVRLARIL  RKCMCESBPOOJKCMEETD                     011600
C         WRAXITVSERNLZONTUZ ATRIL   KVDUNN RRTRFTIRNSOE                     011601
C          DR CAE RLSGEATAGE Y TIE   IIEVCG S AKFAFYGSRP                     011602
C  ADT                                                *                      011603
C  ADDT                                               *                      011604
C  ASIMLR                   **                      * *  **                  011605
C  EABT                                 *                                    011606
C  EAKSLV                   *                                                011607
C  ECDTQP  *     *          *             *   **    *    *                   011608
C  RFGENR  *     *          *             *   **    *                        011609
```

```
C   BSGENR   *     *                      *   *     *               011610
C   CENTRL   **              *                      *  **           011611
C   CREA                                      *       *             011612
C   CREADC                                    *       *             011613
C   CREB                                      *       *             011614
C   CREC                                      *       *             011615
C   CREE                                      *       *             011616
C   CRENUD                                    *       *             011617
C   CRET3                                     *       *             011618
C   DYNS10     *  *  *                     ** *      ****           011619
C   DYNS20  * **   ***                    **  * *  ** ****          011620
C   DYNS30                                    *     *               011621
C   DYNS40              * *******  *****   *    * *   ***           011622
C   ENGMCM   *       *                    * * *     *   *           011623
C   EQADD    * *             *            *         *   *           011624
C   EXTCR                                 *         *   *           011625
C   FINCT                   ***                     *    *          011626
C   FINCU   **              *             *   *     *  ***          011627
C   GAUSSI      *                                                   011628
C   GETEMP  **        *     *             *   *     *               011629
C   GMISC                                 *         *   *           011630
C   GRVCRC  **   **  *      *             *   *     *   *           011631
C   INVINF                  *                                       011632
C   KFINGE   **                           * *       *               011633
C   LINEAR                  **                   *    * **          011634
C           ABCCDGGHIIIJKLLLLLLLLLLLLLLLLLVMMNNPPPPRSSTTVV          011635
C           NHONKGGALNVIDACBDDIKRSSITVVWWZAIOHUILRSPPJAILI          011636
C           LENAADGNIICLSMODESJACIIIGI2OP4ASMNMNT*TRLMPMCN          011637
C           ESPCTAADNGOFISGABIVRCARIL  PKCMCESBPDJJKCALLIG          011638
C           WRAXIIVSERNLZONTUZ ATRIL   KVDUNN RRTRFTIRNSUE          011639
C            GR CAE RLSGLATAGE Y TIL   IIEMCG S ANFAFYGSRP          011640
C   LPLTWR                   *                  *     ***           011641
C   LPRNT                    *                         **           011642
C   LTCFQL              *                              **           011643
C   LTRESP              *    *       * *   *          ***           011644
C   MAIN        **  *                     **  * *   * *             011645
C   MGEN    **    *  *      *             *   *     *   *           011646
C   MRICIC                                   *      ***             011647
C   MSMCDC                                *  **     ***             011648
C   MSMCDL                                   **     ***             011649
C   MULTH                               *                           011650
C   NIPLOT                       **      *        *                 011651
C   NUMS        *                                                   011652
C   NYPLOT                       **      *        *                 011653
C   PAGEHD                       **                                 011654
C   PLOTWR               *                * *   *   * ***           011655
C   PLTCAR                       **      *                          011656
C   PRNTCS  **           *                ***       *  **           011657
C   QHDRVR                  *          *                            011658
C   RKACAM             *                         * *   ***          011659
C   RECCUS                           *            *                 011660
C   REPLOT                       **      *        *                 011661
C   RETCH    *     *        *                  *    *   *           011662
C   RETGS          *                                *   *           011663
C   RETTR                                     *                     011664
C   SFRE32              *   *        *            *                 011665
C   SHAPE1                                *         *   *           011666
C   SPLOT                        **      *        *                 011667
C   START                   *    **      *                          011668
C   TEPLY                                           *               011669
```

```
C   IFTYPE                  *
C   TORQUE  *    *  *      *              * * *     *   *
C   WTAPE                       **
C   YDOT   **     ***** *  *              * * *     * ***
C          ABCDDGGHIIIJKLLLLLLLLLLLLLLLLLMMMNNPPPPQSSTTVV
C          WHONRGGALNVIDACBDDIRRSSTTVVWWZAIOHUILRSPPJAIEI
C          CDNAADSNITCLSMODESJACTTIGI2ORMXSMNMNTWTREMPMCN
C          ESPCTAADNGDF IBOABIVRLARTL  RKOMCESBPDJJKCMEETD
C          WRAXITVSERNLZDNTOZ ATRTL   KVDUNN RRTRFIIRNSUL
C           OR CAL RLSGEATAGE Y TIE   IIEMOG S AKFAFYUSRP
C
C
C
C
C
C**************************************************************
C
C     COMMON /AMJBW/
C        USED IN:
C                   DYNS20     GETBMB     PRNTOU
C                   FINDU      MGEN
C                   YDOT       GRVGRD
C        PARAMETERS:
C     AMC = INERTIA MATRIX FOR BODY N, AT T=0 COMPUTED VIA DYNS10 STORED
C           ON NTAPE1, THEREAFTER UPDATED IN MGEN
C     EW =  COEFFICIENT MATRIX OF EQUATION II-84, VOL I
C           SUBPARTITION OF WHICH IS INVERTED, COMPUTED IN FINDU
C           AREA ALSO USED IN YDOT AND GETBMB FOR WORK ASSOCIATED WITH
C           COMPUTATION OF LAGRANGE MULTIPLIERS
C
C**************************************************************
C
C     COMMON /BHBSRD/
C        USED IN:
C                 FINDU   GETBMB    ENCMCM
C                 YDOT    MGEN      KHINGE
C                 ROTDH   GRVGRD    CONTRL
C                 BHGENR  BODTQP    PRNTOU
C                 BSGENR  TORQUE    EQADU
C        PARAMETERS:
C     BH  = ARRAYS OF ELEMENTS WHICH DEFINE HINGE POINT KINEMATICS
C           COMPUTED IN BHGENR, SEE PAGE 36, VOL I
C     BS  = ARRAYS OF ELEMENTS WHICH DEFINE SENSOR POINT KINEMATICS
C           COMPUTED IN BSGENR, SEE PAGE 17, VOL II
C     ROL = ROTATION TRANSFORMATIONS RELATING THE BODY AXIS SYSTEM
C           TO THE INERTIAL REFERENCE, COMPUTED IN ROTDH, SEE PAGE 18,
C           VOL 2
C     DOL = VECTOR COMPONENTS FROM INERTIAL REFERENCE TO THE BODY FIXED
C           AXIS SYSTEM, COMPUTED IN ROTDH, SEE PAGE 18, VOL II
C
C**************************************************************
C
C     COMMON /CONPAR/
C        USED IN:
C                   DYNS10     CONTRL
C                   DYNS20     (ANY OTHERS REQUIRED BY USER)
C                   KHINGE
C        PARAMETERS:
C     CNTDTA = INPUT IN DYNS10, IN DYNS20 THE FIRST NDELTA ELEMENTS ARE
C               THE INITIAL CONDITIONS FOR THE NDELTA DIFFERENTIAL
```

```
C                 EQUATIONS DEFINED IN CONTRL. THE REST ARE USER DEFINED.     011730
C                 THE USER MAY USE THIS AREA TO INPUT PARAMETERS NEEDED       011731
C                 IN ANY USER DEFINED SUBROUTINE                              011732
C                                                                             011733
C****************************************************                         011734
C                                                                             011735
C      COMMON /CNAUX/                                                         011736
C         USED IN:                                                            011737
C                        DYNSZO                                               011738
C                        EQAUG                                                011739
C         PARAMETERS:                                                         011740
C      NAUX   = NUMBER OF AUXILIARY EQUATIONS USED TO DEFINE THE SENSOR       011741
C                 AND TORQUE SIGNALS REQUIRED FOR THE STABILITY ANALYSIS      011742
C                 NAUX = NASS+NCTQ (IN EQAUG)                                 011743
C                                                                             011744
C****************************************************                         011745
C                                                                             011746
C      COMMON /CRATIO/                                                        011747
C         USED IN:                                                            011748
C                        MAIN                                                 011749
C                        GAUSSI                                               011750
C                        NUMS                                                 011751
C         PARAMETERS:                                                         011752
C      IFL1   = INTEGER FLAG, DEFINES IF CALL TO GAUSSI IS FROM NUMS.         011753
C                 IF SO IFL1=1. INITIALIZE TO ZERO IN MAIN                    011754
C      IFL2   = INTEGER FLAG, IF CALL IS FROM NUMS AND IFL2 SET =1 IN         011755
C                 GAUSSI A ZERO NUMERATOR GAIN CONDITION ENCOUNTERED          011756
C      GRVEC  = PIVOT ELEMENTS USED IN GAUSSI TO COMPUTE A**(-1) THESE        011757
C                 IN TURN TO BE USED TO SET DET(A)                            011758
C                                                                             011759
C****************************************************                         011760
C                                                                             011761
C      COMMON /GGDATA/                                                        011762
C         USED IN:                                                            011763
C                        MAIN                                                 011764
C                        DYNSIO                                               011765
C                        GRVGRD                                               011766
C         PARAMETERS:                                                         011767
C      GAMGI  = INPUT DATA FROM DYNSIO COMPONENTS OF GRAVITY VECTOR IN        011768
C              INERTIAL COORDINATES, SEE SUBROUTINE DEF3  REDEFINED           011769
C                 IN DYNSIO TO BE COMPONENTS OF UNIT VECTOR IN DIRECTION      011770
C                 OF GRAVITY FIELD, EQUIVALENCE(GAMGI(1),WV(1)) IN DYNSIO     011771
C                 (FROM GRAVITY SOURCE TO VICINITY OF MODEL CENTER OF MASS)   011772
C      CMAG   = MAGNITUDE OF LOCAL GRAVITATIONAL ACCELERATION, COMPUTED       011773
C              IN DYNSIO, SEE PAGE 46, VOL I (MAG. OF INPUT GAMGI)            011774
C      FCMAG  = MAGNITUDE OF VECTOR FROM GRAVITY SOURCE TO VICINITY OF        011775
C                 SPACECRAFT, INPUT DATA FROM DYNSIO                          011776
C                                                                             011777
C****************************************************                         011778
C                                                                             011779
C      COMMON /GGSAVE/                                                        011780
C         USED IN:                                                            011781
C                        MGEN                                                 011782
C                        GRVGRD                                               011783
C                        TORQUE                                               011784
C         PARAMETERS:                                                         011785
C      CGS  = GRAVITY GRADIENT EFFECTS ON THE ELASTIC COORDINATES OF EACH     011786
C             BODY, COMPUTED IN MGEN SEE PAGE 52,VOL I                        011787
C                                                                             011788
C****************************************************                         011789
```

```
C                                                                       C11790
C      COMMON /FANDS/                                                   011791
C         USED IN:                                                      011792
C                      DYNS20     BHGENR     BDCTQP                     011793
C                      YDOT       ROTDS                                 011794
C                      ROTOR      BSGENR                                C11795
C      PARAMETERS:                                                      011796
C      FATH = MODAL DISPLACEMENTS AT HINGE POINTS                       011797
C      SICH = MODAL ROTATIONS AT HINGE POINTS                           011798
C      FATS = MODAL DISPLACEMENTS AT SENSOR POINTS                      C11799
C      SICS = MODAL ROTATIONS AT SENSOR POINTS                          011800
C                                                                       011801
C*****************************************************************      C11802
C                                                                       C11803
C      COMMON /ILINER/                                                  C11804
C         USED IN:                                                      011805
C                      MAIN                                             011806
C                      DYNS10                                           011807
C                      DYNS20                                           C11808
C                      YDOT                                             C11809
C         PARAMETERS:                                                   011810
C                                                                       C11811
C      IFLNER = 1 PERFORM LINEAR TIME OR FREQUENCY RESPONSE ANALYSIS    011812
C             = 0 PERFORM NON-LINEAR TIME RESPONSE ANALYSIS             011813
C               THIS INTEGER IS SET IN DYNS10 BY DATA INPUT PARAMETER   011814
C               IPDATA(3), SEE SUBROUTINE DEFS                          011815
C                                                                       011816
C*****************************************************************      C11817
C                                                                       011818
C      COMMON /INTGRL/                                                  C11819
C         USED IN:                                                      C11820
C                      DYNS20     GRVGRD                                C11821
C                      YDOT       TORQUE                                011822
C                      MGEN       ENGMOM                                011823
C      PARAMETERS:                                                      C11824
C            AM = UNDEFORMED INERTIA MATRIX (M SUB 0 IN EQ II-87)       011825
C          ACOF = ALPHA COEFFICIENTS OF EQ II-88 USED TO DEFINE BODY'S  011826
C               STATIC MASS MOMENTS LINEAR DEPENDENCE ON DEFORMATION    011827
C          BCOF = B COEFFICIENTS OF EQ II-88 USED TO DEFINE BODY INERTIA 011828
C               TENSOR'S LINEAR DEPENDENCE ON DEFORMATION               011829
C         COF11 = C SUB 11 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP  011830
C         COF22 = C SUB 22 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP  011831
C         COF33 = C SUB 33 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP  011832
C            AK = MODAL STIFFNESS MATRIX                                011833
C         COF12 = C SUB 12 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP  011834
C         COF13 = C SUB 13 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP  011835
C         COF23 = C SUB 23 COEFFICIENTS OF EQ II-89,INERTIA TENSOR QUAD DEP  011836
C            AD = MODAL DAMPING MATRIX                                  011837
C         COFXY = C SUB XY COEFFICIENTS OF EQ II-85                     011838
C         COFXZ = C SUB XZ COEFFICIENTS OF EQ II-85                     011839
C         COFYZ = C SUB ZY COEFFICIENTS OF EQ II-85                     011840
C                                                                       C11841
C*****************************************************************      011842
C                                                                       C11843
C      COMMON /IVCONS/                                                  C11844
C         USED IN:                                                      011845
C                      YDOT                                             011846
C                      GETBMB                                           C11847
C         PARAMETERS:                                                   011848
C      IV = INTEGER ARRAY SET UP IN GETBMB, USED TO PERFORM THE SELECTIVE  C11849
```

```
C           ROW MULTIPLICATIONS REQUIRED IN GETBMB, PICKS OUT CONSTRAINED 011850
C           AND UNCONSTRIANED COORDINATES AT EACH HINGE POINT.            011851
C           PROVIDES THE NUMBERING SEQUENCE FOR THE LAGRANGE MULTIPLIERS  011852
C           EVALUATED IN YDOT                                             011853
C                                                                         011854
C**************************************************************           011855
C                                                                         011856
C     COMMON /JILFLG/                                                     011857
C        USED IN:                                                         011858
C                    YDOT                                                 011859
C                    RKADAM                                               011860
C        PARAMETERS:                                                      011861
C     JIL = 1,2,3 OR 4 DEFINE CYCLE NUMBER IN FOUR STEP RUNGE KUTTA       011862
C           INTEGRATION ROUTINE. USED IN YDOT TO SET FLAG WHICH SIGNALS   011863
C           WHETHER OR NOT A NEW SET OF INDEPENDENT COORDINATES SHOULD    011864
C           BE EVALUATED IN FINDD                                         011865
C                                                                         011866
C**************************************************************           011867
C                                                                         011868
C     COMMON /KDSIZE/                                                     011869
C        USED IN:                                                         011870
C                    DYNS40     ETCRQL                                    011871
C                    SFRCQ2     ETRESP                                    011872
C        PARAMETERS:                                                      011873
C     KR   = DIMENSION OF ARRAYS IN /LIJV/,/LRARAY/,/LV1/,/LV2/,/LWORK1/  011874
C     KV1  = DIMENSION OF ARRAYS IN /LIJV/,/LRARAY/,/LV1/,/LV2/,/LWORK1/  011875
C     KRT  = DIMENSION OF ARRAY R  IN /LPLOT/                             011876
C     KRX  = DIMENSION OF ARRAY RX IN /LRCOT/                             011877
C     KV2  = DIMENSION OF ARRAY R  IN /LRCOT/                             011878
C     KVX  = DIMENSION OF ARRAY RX IN /LRCOT/                             011879
C                                                                         011880
C**************************************************************           011881
C                                                                         011882
C     COMMON /LAMBDA/                                                     011883
C        USED IN:                                                         011884
C                    YDOT                                                 011885
C                    PRNTGD                                               011886
C                    PLOTWR                                               011887
C        PARAMETERS:                                                      011888
C     ALAM = IN YDOT EQUIVALENCE (ALAM(1),V(1))  AREA USED TO COMPUTE     011889
C             LAGRANGE MULTIPLIERS. UPON EXIT FROM YDOT, ALAM ARRAY       011890
C             CONTAINS THE NEW INTERCONNECTION CONSTRAINT FORCES          011891
C             (LAMBDAS)                                                   011892
C                                                                         011893
C**************************************************************           011894
C                                                                         011895
C     COMMON /LBDATA/                                                     011896
C        USED IN:                                                         011897
C                    DYNS40                                               011898
C        PARAMETERS:                                                      011899
C     FMIN = FREQUENCY SWEEP LOWER LIMIT                       (DYNS40)   011900
C     FMAX = FREQUENCY SWEEP UPPER LIMIT                       (DYNS40)   011901
C     DBMIN= MINIMUM DB AMPLITUDE FOR BODE, NICHOLS PLOTS      (DYNS40)   011902
C     DBMAX= MAXIMUM DB AMPLITUDE FOR BODE, NICHOLS PLOTS      (DYNS40)   011903
C                                                                         011904
C**************************************************************           011905
C                                                                         011906
C     COMMON /LCOUNT/                                                     011907
C        USED IN:                                                         011908
C                    DYNS40                                               011909
```

```
C          PARAMETERS:
C       NNR = NUMBER OF REAL AND ZERO ROOTS IN NUMERATOR   (DCQRRT)
C       ICN = NUMBER OF COMPLEX PAIRS IN NUMERATOR         (DCQRRT)
C       NNZ = NUMBER OF ZEROS IN NUMERATOR                 (DCQRRT)
C       NDR = NUMBER OF REAL AND ZERO ROOTS IN DENOMINATOR (DCQRRT)
C       ICD = NUMBER OF COMPLEX PAIRS IN DENOMINATOR       (DCQRRT)
C       NDZ = NUMBER OF ZEROS IN DENOMINATOR               (DCQRRT)
C
C
C
C
C************************************************************
C
C       COMMON /ZDEBUG/
C          USED IN:
C                    (ALL ROUTINES HAVING DEBUG PRINTING CODED)
C          PARAMETERS:
C       DEBUG  =LOGICAL ARRAY SKIP DEBUG PRINT IN SUBROUTINE (DEBUG # K)
C               IF DEBUG(K) =.FALSE., PRINT IF DEBUG(K) = .TRUE.
C
C************************************************************
C
C       COMMON /ZDSIZE/
C          USED IN:
C                    LINEAR    FINDT     LPLTWR    EDADD
C                    DYN340    IFTYPE    LPRNT
C                    ASIMLR    LTRESP    CONTRL
C          PARAMETERS:
C       NXSS    = NUMBER OF PLANT SENSOR SIGNALS, USER DEFINED IN
C                 CONTRL, DEFAULT IS 1 (FOR STABILITY ANALYSIS)
C       NBTQ    = NUMBER OF TORQUE SIGNALS, USER DEFINED IN
C                 IN CONTRL, DEFAULT IS 0 HOWEVER FOR STABILITY ANALYSIS
C                 NBTQ MUST BE AT LEAST ONE, NBTQ = NB IN SOME ROUTINES
C       NDELTA  = NUMBER OF DIFFERENTIAL EQUATIONS DEFINED IN CONTRL
C                 EQUAL TO INPUT PARAMETER NDELTA IN CONTRL
C       NX      = NUMBER OF INDEPENDENT COORDINATES IN THE STATE VECTOR Y
C                 COMPUTED IN LINEAR
C       NDG     = NX + NUMBER OF AUXILIARY EQUATIONS, COMPUTED IN LINEAR
C       NY2     = NX - NDLTA - NXSS, COMPUTED IN DYN340
C       ND2     = NDLTA - NBTQ      , COMPUTED IN DYN340 (NBTQ=NB)
C       NY      = UNUSED
C
C************************************************************
C
C       COMMON /ZIJV/
C          USED IN:
C                   DYN340
C                   FINDT
C          PARAMETERS:
C       IV = INTEGER WORK VECTOR PASSED TO ASIMLR VIA DYN340, USED THERE
C            FOR REORDERING R**(-1)*H*R. EXIT FINDT WITH IV BEING THE
C            TRANSFORMED STATE VECTOR CORRELATION ARRAY
C       JV = INTEGER WORK AREA USED IN FINDT
C
C************************************************************
C
C       COMMON /ZRARAY/
C          USED IN:
C                   DYN340
C          PARAMETERS:
```

```
C      FTRN = TIME CONSTANTS ASSOCIATED WITH ALL REAL ROOTS OF NUMERATOR
C      FRNC = FREQUENCIES OMEGA AND DAMPING ZETA'S ASSOCIATED WITH ALL
C             COMPLEX ROOTS OF THE NUMERATOR
C             FORM:
C                    S**2 + 2*ZETA*OMEGA*S + OMEGA**2 = 0
C      FTRD = TIME CONSTANTS ASSOCIATED WITH ALL REAL ROOTS OF DENOMIATOR
C      FRDC = FREQUENCIES OMEGA AND DAMPING ZETA'S ASSOCIATED WITH ALL
C             COMPLEX ROOTS OF THE DENOMINATOR
C      ** NOTE ** NUMERATOR/DENOMINATOR = TRANSFER FUNCTION REQUESTED
C
C**************************************************************
C
C      COMMON /LROOT/
C         USED IN:
C                    DYNS40
C         PARAMETERS:
C      R  = ROOTS ARRAY COMPUTED IN TIFF FOR PARTICULAR TRANSFER FUNCTION
C           ROOTS COMMON TO DENOMINATOR AND NUMERATOR CANCELLED IN CANCOR
C      RR = ARRAY USED TO STORE ALL COEFFICIENTS OF THE NUMERATOR AND
C           DENOMINATOR POLYNOMIALS, ROOTS IN R USED TO COMPUTE THESE
C           COEFFICIENTS IN RTCP
C
C**************************************************************
C
C      COMMON /LSTART/
C         USED IN:
C                    START      SPLOT      REPLOT
C                    PAGEHD     NIPLOT     PLTCAR
C                    WTAPE      KYPLOT
C         PARAMETERS:
C      IRUNNO = RUN NUMBER, INPUT IN START
C      IDATE  = DATA OF RUN FROM LIBRARY ROUTINE
C      NPAGE  = PAGE COUNTER UPDATED IN PAGEHD
C
C**************************************************************
C
C      COMMON /LSTRT1/
C         USED IN:
C                    START      SPLOT      REPLOT
C                    PAGEHD     NIPLOT     PLTCAR
C                    WTAPE      KYPLOT
C         PARAMETERS:
C      UNAME  = USERS NAME, INPUT START
C      TITLE1 = FIRST TITLE EACH PAGE, INPUT START
C      TITLE2 = SECOND TITLE EACH PAGE, INPUT START
C
C**************************************************************
C
C      COMMON /LTITLE/
C         USED IN:
C                    DYNS40
C         PARAMETERS:
C      TITLE = INPUT FORMAT 20A4, PUT ON ALL FREQUENCY DOMAIN OUTPUT DATA
C              PRINT-OUT AND PLOTS
C
C**************************************************************
C
C      COMMON /LTOL/
C         USED IN:
C                    DYNS40
```

```
C          PARAMETERS:                                                  012030
C      TOLN , TOLD = TOLERANCE PARAMETERS ALL NUMERATOR AND DENOMINATOR 012031
C                    ROOT LESS THAN TOLN,TOLD ARE SET TO ZERO IN CALL TO 012032
C                    SIFT VIA DYNS40                                    012033
C                                                                       012034
C************************************************************          012035
C                                                                       012036
C      COMMON /LV1/                                                     012037
C         USED IN:                                                      012038
C                    DYNS40     REDUCS                                  012039
C                    SFREQZ     LTRESP                                  012040
C         PARAMETERS:                                                   012041
C      V1,V2,V3 = TEMPORARY WORK AREAS                                  012042
C                                                                       012043
C************************************************************          012044
C                                                                       012045
C      COMMON /LV2/                                                     012046
C         USED IN:                                                      012047
C                    DYNS40                                             012048
C         PARAMETERS:                                                   012049
C      XV1 = WORK AREA USED IN DYNS40 FOR FREQUENCY RESPONSE ANALYSIS   012050
C      XV2 = WORK AREA USED IN DYNS40 FOR FREQUENCY RESPONSE ANALYSIS   012051
C      XV3 = WORK AREA USED IN DYNS40 FOR FREQUENCY RESPONSE ANALYSIS   012052
C      XV4 = WORK AREA USED IN DYNS40 FOR FREQUENCY RESPONSE ANALYSIS   012053
C                                                                       012054
C************************************************************          012055
C                                                                       012056
C      COMMON /LWORK1/                                                  012057
C         USED IN:                                                      012058
C                    DYNS40                                             012059
C                    GROFVR                                             012060
C                    LTRESP                                             012061
C         PARAMETERS:                                                   012062
C      W1 = USED AS WORK AREAS                                          012063
C      W2 = USED AS WORK AREAS                                          012064
C                                                                       012065
C************************************************************          012066
C                                                                       012067
C      COMMON /LWRKV1/                                                  012068
C         USED IN:                                                      012069
C                    BABT                                               012070
C                    MULTB                                              012071
C         PARAMETERS:                                                   012072
C      W = WORK AREA                                                    012073
C                                                                       012074
C************************************************************          012075
C                                                                       012076
C      COMMON /LZMODE/                                                  012077
C         USED IN:                                                      012078
C                    START      NIPLOT                                  012079
C                    PLTCAR     NYPLOT                                  012080
C                    SPLOT      RLPLOT                                  012081
C         PARAMETERS:                                                   012082
C      AMODE  = ARRAY SET UP BY SC4020 ROUTINE MODESG                   012083
C                                                                       012084
C************************************************************          012085
C                                                                       012086
C      COMMON /MAXMUM/                                                  012087
C         USED IN:                                                      012088
C                    MAIN       YOUT       GRVGRD     KHINGE     PRNTOU 012089
```

```
C                   DYNS10     BRGENR     BOUTOP     EXTOR                 012090
C                   MSMODC     ESGENR     TORQUE     SHAFFT                012091
C                   DYNS20     GETBNB     ENGMOM     EQADJ                 012092
C                   FINDU      MGEN       PLOTWR     GMISC                 012093
C        PARAMETERS:                                                       012094
C     NBMAX  = MAXIMUM NUMBER OF BODIES                                    012095
C     NHMAX  = MAXIMUM NUMBER OF HINGES                                    012096
C     NSPMAX = MAXIMUM NUMBER OF SENSOR POINTS                             012097
C     NMWMAX = MAXIMUM NUMBER OF MOMENTUM WHEELS                           012098
C     NMWBOD = MAXIMUM NUMBER OF MOMENTUM WHEELS PER BODY                  012099
C     NMCMOD = MAXIMUM NUMBER OF FLEXIBLE BODY MODES PER BODY              012100
C     KMU    = MAXIMUM NUMBER OF VELOCITY VARIABLES PER BODY (U'S)         012101
C     KY     = MAXIMUM NUMBER OF PARAMETERS IN THE STATE VECTOR            012102
C     KU     = MAXIMUM NUMBER OF VELOCITY VARIABLES FOR SYSTEM             012103
C         (ALL OF THESE PARAMETERS ARE SET IN MAIN. TO CHANGE ANY OR       012104
C           ALL OF THEM THE REDIMENSION PROGRAM MUST BE USED WHICH         012105
C           WILL REDIMENSION THE APPROPRIATE PARAMETERS IN THE PROGRAM     012106
C           REDIMENSIONED PROGRAM MUST THEN BE RECOMPILED)                 012107
C                                                                          012108
C*********************************************************************     012109
C                                                                          012110
C     COMMON /MISCNO/                                                      012111
C        USED IN:                                                          012112
C                   MAIN       PRNTOU                                      012113
C                   DYNS10     LTRESP                                      012114
C                   DYNS20     DYNS40                                      012115
C        PARAMETERS:                                                       012116
C     NOPRNT = INPUT PARAMETER IPDATA(1) READ IN DYNS10, PRINT OUTPUT      012117
C              DATA EVERY NOPRNT MULTIPLES OF INTEGRATION STEP             012118
C     NOPLOT = INPUT PARAMETER IPDATA(2) READ IN DYNS10, PLOT CONTROL      012119
C             FOR TIME RESPONSE PLOT EVERY NOPLOT MULTIPLES OF             012120
C              INTEGRATION STEP                                            012121
C                                                                          012122
C                                                                          012123
C                                                                          012124
C*********************************************************************     012125
C                                                                          012126
C     COMMON /MOMENG/                                                      012127
C        USED IN:                                                          012128
C                   YDOT       PRNTOU                                      012129
C                   TORQUE     PLOTWR                                      012130
C                   ENGMOM     KHINGE                                      012131
C        PARAMETERS:                                                       012132
C     H      = ORDINARY MOMENTA COMPUTED IN YDOT, SEE EQUATION             012133
C             II-44,VOL I                                                  012134
C     HMOM   = CONTRIBUTION TO TOTAL ANGULAR AND LINEAR MOMENTUM OF        012135
C             EACH BODY, COMPUTED IN ENGMOM                                012136
C     HTOT   = TOTAL SYSTEM ANGULAR MOMENTUM VECTOR, COMPUTED IN ENGMCM    012137
C     PTOTAL = TOTAL SYSTEM LINEAR MOMENTUM VECTOR, COMPUTED IN ENGMOM     012138
C     ENGKE  = CONTRIBUTION TO TOTAL KINETIC ENERGY OF EACH BODY,          012139
C              COMPUTED IN ENGMOM                                          012140
C     ENGPE  = CONTRIBUTION TO TOTAL POTENTIAL ENERGY OF EACH BODY,        012141
C              COMPUTED IN ENGMOM                                          012142
C     TOTKE  = TOTAL KINETIC ENERGY      COMPUTED IN ENGMOM                012143
C     TOTPE  = TOTAL POTENTIAL ENERGY, COMPUTED IN ENGMOM                  012144
C     TOTENG = TOTAL ENERGY (T+V)      , COMPUTED IN ENGMOM                012145
C     AMTOT  = TOTAL ANGULAR MOMENTUM  , COMPUTED IN ENGMOM                012146
C     ALTOT  = TOTAL LINEAR MOMENTUM   , COMPUTED IN ENGMOM                012147
C                                                                          012148
C*********************************************************************     012149
```

```
C
C     COMMON /NHNS/
C        USED IN:
C                    DYNS10     MRIGID
C                    MSMODL
C                    MSMODC
C        PARAMETERS:
C     NHBOI  = NUMBER OF HINGES ON EACH BODY, COMPUTED IN DYNS10
C     NSBOI  = NUMBER OF SENSOR POINTS ON EACH BODY, COMPUTED IN DYNS10
C
C*****************************************************************
C
C     COMMON /NUMBRS/
C        USED IN:
C                    MAIN       CRLA       FINDD      GETJMB     ENGMCM
C                    MSMODC     CRLB       YDOT       MSEN       CREMOD
C                    MSMODL     CREC       CRLE       GRVGRD
C                    CRETJ      ROTTR      BHGENR     BDJTGP
C                    CREADD     DYNS20     BSGENR     TORQUE
C        PARAMETERS:
C     ZRO  =  0.D0    (ALL SET IN MAIN)
C     ONE  =  1.D0
C     TWO  =  2.D0
C     THRE =  3.D0
C
C*****************************************************************
C
C     COMMON /FINPF/
C        USED IN:
C                    ROTDH
C                    BHGENR
C                    BDCTGP
C        PARAMETERS:
C     FIN = ROTATION TRANSFORMATION MATRIX RELATING BODY FIXED ANGULAR
C           RATES TO EULER RATES BETWEEN P AND Q FRAME AT EACH HINGE,
C           COMPUTED IN ROTDH
C     RP2 = COMPUTED IN ROTTR VIA CALL FROM ROTDH
C     RP3 = COMPUTED IN ROTTR VIA CALL FROM ROTDH
C           (BOTH RP2,RP3 USED IN BDCTGP TO OBTAIN D/DT(PI(INVERSE))
C
C*****************************************************************
C
C     COMMON /FLTDTA/
C        USED IN:
C                    MAIN       DYNS30
C                    DYNS20     DYNS40
C                    PLOTWR     LPLTWR
C        PARAMETERS:
C     NRFLCT = BOTH ARE INTEGERS COMPUTED IN PLOTWR, LPLTWR AND DYNS30
C     NCFLCT   THEY ARE USE IN DYNS30 FOR CREATION OF PLOTS
C
C*****************************************************************
C
C     COMMON /FRWORK/
C        USED IN:
C                    LINEAR
C                    RKADAM
C        PARAMETERS:
C     FR = DUAL PURPOSE WORK AREA, TEMPORARY STORAGE AREA FOR ADAMS P/C
C           INTEGRATION ALGORITHM IN RKADAM, TEMPORARY STORAGE AREA IN
```

```
C           LINEAR FOR OBTAINING PARTIAL DERIVATIVE MATRIX                012210
C                                                                        012211
C************************************************************            012212
C                                                                        012213
C      COMMON /ZFSTOFF/                                                  012214
C         USED IN:                                                       012215
C                       SFREQ2     A1PLOT     DYNS40                     012216
C                       RLOCUS     NYPLOT                                012217
C                       SPLOT      REPLOT                                012218
C        PARAMETERS:                                                     012219
C      KSAVE    TOTAL NUMBER OF POINTS SAVED FOR THE PLOTTING OF THE     012220
C               FREQUENCY RESPONSE DATA                                  012221
C      SAVEF    (IN SFREQ2) THE ARRAY OF FREQUENCIES AT WHICH THE TRANSFER 012222
C               FUNCTION IS EVALUATED THAT IS ALL OMEGA FOR WHICH        012223
C                     TF(I*OMEGA) = ALPHA + I*BETA                       012224
C               IS COMPUTED                                              012225
C               (IN RLOCUS) EQUIVALENCE (SAVEF(1),XSAVE(1)) THE ARRAY    012226
C               CONTAINES THE REAL COORDINATES OF THE POINTS FOUND TO LIE 012227
C               ALONG THE ROOT LOCUS                                     012228
C      SAVEP    (IN RLOCUS) EQUIVALENCE (SAVEP(1),YSAVE(1)) THE ARRAY    012229
C               CONTAINES THE IMAGINARY COORDINATES OF THE POINTS FOUND  012230
C               TO LIE ALONG THE ROOT LOCUS                              012231
C               (IN SPLOT2)                                              012232
C                       AR = DSQRT(ALPHA**2 + BETA**2)                   012233
C                     DPHI = DATAN2(BETA,ALPHA)*57.2958                  012234
C                          0 < DPHI < 360.0                              012235
C                   DBEL = 8.685889641*DLOG(AR)  WHEN  8.6859 = 20*LOG(E) 012236
C      SAVEG  = DBEL  FOR EACH OMEGA                                     012237
C      SAVEF  = DPHI  FOR EACH OMEGA                                     012238
C      SAVEA  = AR    FOR EACH OMEGA                                     012239
C                                                                        012240
C                                                                        012241
C                                                                        012242
C************************************************************            012243
C                                                                        012244
C      COMMON /ZRKTA/                                                    012245
C         USED IN:                                                       012246
C                       DYNS20                                           012247
C                       RKADAM                                           012248
C        PARAMETERS:                                                     012249
C      DRK    = INITIALIZED IN DYNS20 TO 0.0D0. TEMPORARY STORAGE AREA USED 012250
C               BY NUMERICAL INTEGRATION ROUTINE                         012251
C      RRK    = CONSTANTS REQUIRED BY INTEGRATION ALGORITHM, SET IN DYSN20 012252
C      NT     = COUNTER UPDATE AT END OF EACH INTEGRATION STEP           012253
C                                                                        012254
C************************************************************            012255
C                                                                        012256
C      COMMON /ZSPECIF/                                                  012257
C         USED IN:                                                       012258
C                       MAIN      FINDO     MGEN      POSTWR    SHAFFT   012259
C                       DYNS10    YOUT      GRVGRD    ASIMER    EQADD    012260
C                       ASMODE    ENGENR    BOOTQP    FINDT     GMISC    012261
C                       MSMODE    ROTOS     TORQUE    KTINGE    ROTDH    012262
C                       MRIS10    GOGENR    ENGMCM    CONTRL             012263
C                       DYNS20    GETDMR    PRNTOU    EXTOR              012264
C         PARAMETERS:                                                    012265
C      BETAF  = INPUTTED IN DYNS10. CONTIGUOUS HINGE FRAME ORIENTATION   012266
C               INITIAL CONDITIONS. UPDATED RELATIVE ORIENTATION DATA    012267
C               STRIPPED FROM STATE VECTOR Y IN ROTDH. OUTPUTTED         012268
C               IN PRNTOU. SEE PAGE 13,VOL II AND SUBROUTINE DEF3        012269
```

```
C       BETAHD = INPUTTED IN DYNSIO, CONTIGUOUS HINGE FRAME RELATIVE RATE   012270
C                INITIAL CONDITIONS. UPDATED RELATIVE RATE DATA COMPUTED    012271
C                IN YOUT STORED IN YDT AND BETAHD ARRAYS. SEE PAGES         012272
C                16, VOL II AND SUBROUTINE DEF3                             012273
C       WMC    = INPUTTED IN DYNSIO. MOMENTUM WHEEL SPIN RATE AND SPIN      012274
C                INERTIA INITIAL CONDITIONS. VALUES NOT UPDATED. SEE        012275
C                SUBROUTINE DEF3                                            012276
C       RH     = INTERMEDIATE ROTATION TRANSFORMATIONS USED TO COMPUTE      012277
C                RELATIVE ORIENTATION OF BODY FIXED REFERENCE FRAMES        012278
C                COMPUTED IN ROTOM, SEE COMMENT CARDS THEREIN               012279
C       RS     = ROTATION TRANSFORMATION BETWEEN SENSOR AXIS SYSTEM AND     012280
C                BODY FIXED AXIS SYSTEM, COMPUTED IN ROTOS, SEE ROTOS ALSO  012281
C                SEE PAGE 19,VOL II. INITIAL VALUES COMPUTED IN ROTTR       012282
C                VIA A CALL FROM MRIGID                                     012283
C       DH     = VECTOR TO POSITION HINGE TRIAD WITH RESPECT TO BODY TRIAD  012284
C                INPUTTED IN MRIGID, IF FLEXIBLE BODY COMPUTED IN MSMODL    012285
C                OR MSMODC, UPDATED IN ROTOM, SEE SUBROUTINE DEF3           012286
C       DS     = VECTOR TO POSITION SENSOR TRIAD WITH RESPECT TO BODY       012287
C                TRIAD INPUTTED IN MRIGID, IF FLEXIBLE BODY COMPUTED IN     012288
C                MSMODL OR MSMODC, UPDATED IN ROTOS, SEE SUBROUTINE DEF3    012289
C       IMC    = BASIC MOMENTUM WHEEL DATA INPUTTED IN DYNSIO, SEE PAGE     012290
C                SUBROUTINE DEF3, DEFINES SENSOR POINT NUMBER FOR WHEEL,    012291
C                SPIN AXIS NUMBER AND WHETHER IT IS ACTIVE OR CONSTANT      012292
C                SPEED, ALSO SEE SUB. DEF4 OUTPUT DYNSIO                    012293
C       NMCW   = THE MOMENTUM WHEEL/BODY TABLE, SEE DEF4 OUTPUT DYNSIO      012294
C                DEFINES NUMBER OF WHEELS ON BODY, WHICH ARE VARIABLE       012295
C                SPEED AND RESPECTIVE SENSOR POINT NUMBERS, COMPUTED IN     012296
C                DYNSIO                                                     012297
C       NPTSBW = SENSOR POINT/BODY CORRELATION ARRAY DEFINES BODY NUMBER    012298
C                ON WHICH EACH SENSOR POINT DEFINED. INPUTTED IN DYNSIO.    012299
C                SEE SUB DEF3 AND DEF4   INPUT/OUTPUT DYNSIO                012300
C       NB     = NUMBER OF BODIES IN MODEL, INPUT DYNSIO, SEE DEF3          012301
C       NH     = NUMBER OF HINGES IN MODEL, INPUT DYNSIO, SEE DEF3          012302
C       NSPT   = NUMBER OF SENSOR POINTS  , INPUT DYNSIO, SEE DEF3          012303
C       NOFMC  = NUMBER OF MOMENTUM WHEELS, INPUT DYNSIO, SEE DEF3          012304
C       NDELTA = NUMBER CONTRL DIFF. EQS. , INPUT DYNSIO, SEE DEF3          012305
C       ITOPOL = TOPOLOGY ARRAY,             , INPUT DYNSIO, SEE DEF3 DEFINES 012306
C                THE BODY NUMBERS OF THE CONTIGUOUS BODIES AT EACH HINGE.   012307
C                ALSO DEFINES DIRECTION OF POSITIVE EULER ANGLE ROTATION    012308
C                              ROT = A1*A2*A3                               012309
C                A1,A2,A3 TRANSFORMATIONS FOR 1-ST,2-ND,3-RD POSITIVE       012310
C                ROTATION, SEE PAGES 4-7, VOL II                            012311
C       IRGFLX = NUMBER OF FLEXIBLE BODY MODES FOR EACH BODY, SEE PAGE      012312
C                4-7, VOL II AND DEF3                                       012313
C       IHDATA = HINGE POINT CONSTRAINT DATA, SEE DEF3, INPUT DYNSIO PP4-7  012314
C                VOL 2                                                      012315
C       LOCU   = STATE VECTOR LOCATION ARRAY, SEE DEF4, COMPUTTED           012316
C                IN DYNSIO                                                  012317
C       LENU   = STATE VECTOR LENGTH ARRAY, SEE DEF4, COMPUTTED IN          012318
C                DYNSIO                                                     012319
C       NU     = NUMBER OF U'S IN STATE VECTOR, SEE DEF4, PP4-7 OF VOL II   012320
C                COMPUTED IN DYNSIO                                         012321
C       NBETA  = NUMBER OF BETA'S IN STATE VECTOR, SEE DEF4 + PP4-7 VOL II  012322
C                COMPUTED IN DYNSIO                                         012323
C       NLAM   = NUMBER OF LAMBDA'S(CONSTRAINTS),SEE DEF4 + PP4-7 VOL II    012324
C                COMPUTED IN DYNSIO                                         012325
C       NEQ    = NUMBER OF STATE EQUATIONS TO INTEGRATE SEE DEF4            012326
C                COMPUTED IN DYNSIO                                         012327
C                                                                           012328
C**********************************************************************     012329
```

```
C
C     COMMON /SUMMRY/
C        USED IN:
C                   DYNS10     MRIGID
C                   MSMODE
C                   MSMODE
C        PARAMETERS:
C     KSUMRY = ROW DIMENSION OF ASUMRY AND ISUMRY ARRAYS (DYNS10)
C     ASUMRY = REAL NUMBER WORK AREA ARRAY
C     ISUMRY = INTEGER NUMBER WORK AREA ARRAY
C
C*****************************************************************
C
C     COMMON /TAPENO/
C        USED IN:
C                   MAIN       CREADD     DYNS20     DYNS40     CREE
C                   DYNS10     CREA       YOUT       ASIMLR
C                   MSMODE     CREC       PLOTWR     LPLTWR
C                   MSMODE     CRED       LINEAR     LTRESP
C                   CRET3      MRIGID     DYNS30     CREMOD
C        PARAMETERS:
C     NTAPE1 = 1 SCRATCH TAPE NUMBER SET IN MAIN
C     NTAPE2 = 2 SCRATCH TAPE NUMBER SET IN MAIN
C     NTAPE3 = 3 SCRATCH TAPE NUMBER SET IN MAIN
C
C*****************************************************************
C
C     COMMON /TIMESS/
C        USED IN:
C                   DYNS10     PRNTOU     LPRNT
C                   DYNS20     PLOTWR     LTORQL
C                   FINDD      DYNS40     CENTRL
C                   YOUT       LTRESP
C                   RKADAM     LPLTWR
C        PARAMETERS:
C     ST    = STARTT = TADATA(1) INPUTTED PARAMETER DYNS10 (INITIAL TIME)
C     DT    = DELTAT = TADATA(2) INPUTTED PARAMETER DYNS10 (STEP SIZE)
C     T     = TIME PARAMETER INCREMENTED IN RKADAM OR LTRESP
C     ET    = ENDT = TADATA(3) INPUTTED PARAMETER DYNS10 (END RUN TIME)
C     TMST  = USED IN DYNS20, TIME ELAPSED FROM STARTT INCREMENTED RKADAM
C             OR LTRESP
C
C*****************************************************************
C
C     COMMON /VECTOR/
C        USED IN:
C                   DYNS20     GRVGRD     DYNS40     ADT
C                   FINDD      MOCTSP     ASIMLR     ADDT
C                   YOUT       TORQUE     TFPLY      CONTRL
C                   LINEAR     ENGMOM     LTRESP     EXTOR
C                   ROTOR      RKADAM     LPLTWR     STARTT
C                   ROTOS      PRNTOU     LPRNT      EXADD
C                   MOEN       PLOTWR     LTORQL     GAISC
C        PARAMETERS:
C     Y  = SYSTEM STATE VECTOR (SEE APPENDIX F, VOL I FOR ARRANGEMENT)
C     YD = FIRST TIME DERIVATIVE OF SYSTEM STATE VECTOR, ALL EQUATIONS
C          TO BE INTEGRATED INCLUDED. IN EQADD MAJR AUXILIARY EQUATIONS
C          ADDED FOR LINEAR STABILITY ANALYSIS. (YDT = YD)
C
C*****************************************************************
```

```
C
C     COMMON /VINDEP/
C        USED IN:
C                 DYNS20    RKADAM
C                 FINDU     ASIMLR
C                 LINEAR    FINDT
C        PARAMETERS:
C     INDEF  = USED TO DEFINE INDEPENDENT COORDINATES IN THE STATE
C              VECTOR Y, FROM FINDU IF INDEP(I) = 0 COORDINATE I
C              IS DEPENDENT IF = 1 IT IS INDEPENDENT. IN FINDU A OPTIMUM
C              SET OF INDEPENDENT COORDINATES IS EVALUATED AT THE END
C              OF EACH FULL INTEGRATION STEP. IN DYNS20 ALL COORDINATES
C              IN STATE VECTOR Y INITIALLY TAKEN AS INDEPENDENT
C     IV     = IN LINEAR EQUIVALENCE(INDEP(1),IV(1))
C
C**************************************************************
C
C
C
C
      RETURN
      END
```

```
      SUBROUTINE DEF2
C                                                    DEBUG # 113
C
C
C
C
C
C
C
C                 SUBROUTINE CALLED BY/SUBROUTINE CALLS
C                      CROSS-REFERENCE MATRIX
C
C        ABBBBCCCCCCCCCCCDDDDEFFGGIILLLLLLMMMMMNNNPPPQRRRRRRRRRRRSSSTTTWWY
C        SACHSCORRRRRRRYYYYNIIERNNIPTTTAGRSSIUYALRRREEKLLJJOTWFPTFFORRD
C        IKCGGMNEEEEEEENNNNGNNTVIVNRACRIEIMMPMPGINCDAAAGPIITARRLAPTRIIO
C        MSTEEETAAGEEMTSSSSMDDDGAIENPRENNGUOLSEEEIURJDDDEDDTPIECRLYGTTT
C        LLGNNNR D    U312340IGMRPNATECS  IDDO O4AGNV IAJGRGRETQTTYPUEI
C        RVPRRTL C    C 0000M  BDEPR  LP  DCLT TJRJ R MMST    E2   EE M
C  ACDT                                                                  *
C  ALC3             **
C  ACT                        *
C  ALPHAA     (NONE)
C  ASIMLR                  *
C  ASGR                    *
C  BABT       (NONE)
C  BAKSLV                                                                *
C  BCOTGF                                                                *
C  BFGENR                *                                               *
C  BSGENR                                                                *
C  BTABA                                    *
C  CANCOR                  *
C  CCMENT                                *
```

```
C  CENTOL                                                           *      012444
C  CHLA                                     *                              012445
C  CMPAXL                                   *                              012446
C  CROC                                     *                              012447
C  CRPR                                     *                              012448
C  CRST                                     *                              012449
C  CSTMJ                                    *                              012450
C  CYCL3                                    *                              012451
C  DCCYT                                                               *   012452
C  DECOST                  *                                               012453
C  DEF1       (NONE)                                                       012454
C  DEF2       (NONE)                                                       012455
C  DERK13                  *                                               012456
C  DIPAS                                                *      *           012457
C  DISAG                                                *      *           012458
C  DYNS10                               *                                  012459
C  DYNS20                               *                                  012460
C  DYNS30                               *                                  012461
C  DYNS40                               *                                  012462
C        ABCDBCCCCCCCCCCCDDDDEFFGGIIILLLLLLMMMMNNNPPPQQRRRRRRRRRRRSSSTTTWWY   012463
C        SADHSECDRRRDRRYYYYIIIERNNIPTTTAGRSSIUYALRRRECKLLJJDGTWFPTFFCRRD   012464
C        IKGGCMNEELEEEEENNNNGNNIVIVNRACRIEIMMPMPGTNCDAAADPTTTARRLAPTRIIO   012465
C        MSTICETIAACHMTSSSSADDEGAIENPREMNGOCESLECTORGGDCLJJTPIEDRLYGTTT   012466
C        EECANNE D   G3123AUIGMRPNAIEGS  IDDD DIAGNW IADDHSRETQTTYPGEI   012467
C        SVPOKIL G    C JJJJM  CDEPR  LP  DCET TDRG K 4MS4     E2   EE M   012468
C  EIGVEC                  *                                               012469
C  ENINV                 *                                                 012470
C  FMAGD                                                            *      012471
C  FXEQR                                                            *      012472
C  FETCH        *******                     *                              012473
C  FINIT *                                                                 012474
C  FINDU                 *                                             *   012475
C  FIT                                                  *                  012476
C  FORMAT     (NONE)                                                       012477
C  GAUSSI*                             *   *  *                            012478
C  GETTAG                                                              *   012479
C  GMISC                                                            *      012480
C  GRVCRD                                                              *   012481
C  INTAPE                                            *                     012482
C  INVINP                                         *                    *   012483
C  KFINGE                                                           *      012484
C  LINEAR                *                                                 012485
C  LPLTWP                              *                                   012486
C  LPRNT                               *                                   012487
C  LTAPE                                             *       *             012488
C  LTCRGL                              *                                   012489
C  LTRESP                  *                                               012490
C  MAIN       (NONE MAIN PROGRAM)                                          012491
C  MCEN                                                                *   012492
C  MLTSR                       *                                           012493
C  MRIGID               *                                                  012494
C  MSMCOC               *                                                  012495
C  MSMCOL               *                                                  012496
C  MULTA *                                 *                               012497
C  MULTAD  *               *    *        *                          *      012498
C  MULTB *                             *                                   012499
C  MULT3   ***             *    *        *                ***       *  *   012500
C        AELDBCCCCCCCCCCCDDDDEFFGGIILLLLLLMMMMANNNPPPQQRRRRRRRRRRRSSSTTTWWY   012501
C        SADHSCDRRRRRRRRYYYYYNIIERNNIPTTTAGRSSIUYALRRREEKLLJJDGTWFPTFFCRRD   012502
C        IKGGCMNEELEEEEENNNNGNNIVIVNRACRIEIMMPMPGTNCDAAADPTTTARRLAPTRIIO   012503
```

```
C        MSTEECTAABCEMTSSSSMDDDBGAIENPRENNGOOLSLEETORDDDCLDDTPIEORLYCTTT  012504
C        LLGNNNR D   U31234UTJMRPNATEGS  IDDD DHAUNV IAJUHSRETQTTYPUEI    012505
C        RVPRRTL D    D DDDCM  DDLPR  LP  DCLT TDRD R MMST     E2   EE M  012506
C  NIPLOT                 *                                               012507
C  NUMS                   *                                               012508
C  NYPLOT                 *                                               012509
C  PAGEHD     *        * ** *   *  **    **      *  ** *     **  *  **    012510
C  PLOTSS                                     *         *                 012511
C  PLOTWR               *                                                 012512
C  PLTCAR                *                                                012513
C  PRNTOL               *                                                 012514
C  PR3                                      *                             012515
C  QRCLN                                             *                    012516
C  QRDRVR                 *                  *                            012517
C  QR2                                              *                     012518
C  REAC         *       *  *              ***                             012519
C  REACIM               *  *               *                              012520
C  REVADC*                  *                                             012521
C  REVISE                                   *                             012522
C  RKACAM               *                                                 012523
C  RLOCUS                 *                                               012524
C  RLPLOT                 *                                               012525
C  RMVZRO                 *                                               012526
C  ROTCH                *                                             *   012527
C  ROTCS                                                              *   012528
C  RLTTR                                   ***             **             012529
C  RTAPE                                             *                    012530
C  RTOP                   *                                               012531
C  RWRITE                 *                                               012532
C  SATE          ******                                                   012533
C  SCALE                                                        *         012534
C  SFREQ2                 *                                               012535
C  SHAFTT                                                           *     012536
C  SIFT                   *                  *                            012537
C  SKEWV3  ***                             **                        *    012538
C  SPLOT                  *                                               012539
C  START                                 *                                012540
C  STORE              *                                                   012541
C  SUBDIA                                            *                    012542
C  TFPLY        *                                                         012543
C        AEBBBCCCCCCCCCCDDDDEFFGGIILLLLLLMMMMNNNPPPQQRRRRRRRRRRSSSTTTWWY  012544
C        SADHSCDRRRRRRRRYYYYNIIERNNIPTTTAGRSSIUYAERRREEKLLUUDTWFPTFFERRD  012545
C        IKUJUMNEEEEEEEENNNNGNNIVIVNRACRIEIMMPMPGINCDAAAUPTTTARRLAPTRIIO  012546
C        MSTEECTAABCEMTSSSSMDDDBGAIENPRENNGOOLSLEETORDDDCLDDTPIEORLYCTTT  012547
C        LLGNNNR D   U31234UTJMRPNATEGS  IDDD DHAUNV IAJUHSRETQTTYPUEI    012548
C        RVPRRTL D    D DDDCM  DDLPR  LP  DCLT TDRD R MMST     E2   EE M  012549
C  TFTYPE                 *                                               012550
C  TORQUE                                                             *   012551
C  TRFE       (NONE)                                                      012552
C  TTFF                   *                                               012553
C  UNITY                                    *                             012554
C  WRITE *     *******    *                *  *      *                    012555
C  WRITES****          *    **** ***    ****     * *               *  *   012556
C  WRITIM     (NONE)                                                      012557
C  WRITIS*             **   **           ***             *                012558
C  WTAPE                                             *                    012559
C  YLOT                 *         *                   *                   012560
C  YOCTL                                *                                 012561
C  ZERO  *     ** **      * *        **  ***                      *       012562
C  DATE                                                        *          012563
```

```
C  SECOND                                        * *                         012564
C  TIME                                          *                           012565
C  BCXG                                       * *         *       *          012566
C  EXITG                                                           *         012567
C  GRIDG                                      *   *       *       *          012568
C  LABELG                                     *   *       *       *          012569
C  LEGNDG                                     * * *       *       *          012570
C  LINESG                                     * * *       *       *          012571
C  MODESG                                                          *         012572
C  NUMERG                                         *                          012573
C  OBJCTG                                     * * *       *       *          012574
C  PAGEG                                      * * *       *       *          012575
C  PGRIDG                                       *                            012576
C  PLABLG                                       *                            012577
C  PLINEG                                       *                            012578
C  PSETG                                        *                            012579
C  PSUBJG                                       *                            012580
C  RSETMG                                     * * *       *       *          C12581
C  SETSMG                                     * * *       *       *          012582
C  SETUPG                                         *       *                  012583
C  SUBJEG                                     * * *       *       *          C12584
C  TEXTG                                      * * *       *       *          012585
C  TITLEG                                         *       *       *          012586
C         AEBEBCCCCCCCCCCDDDDEFFGGIILLLLLLMMMMMNNNPPPQQRRRRRRRRRRRSSSTTTWXY   012587
C         SAOHSCCHRRRRRRRYYYYYNIICRNNIPITTTAGRSSIUYALRRRCCKLLDDDTWFPTFFGRRD   012588
C         IRGGGMREECCCCEELNNNNGNNIVIVNRACRISIMMPMPGINCOAAAOPTTTARRLAPTRIIO   C12589
C         MSTCCCTAAOCEMTSSSS4OOBGAICNPRENNGCOLSCCCTGRODDCCDDTPIEGRLYGITT     012590
C         CCDNNN  D    D12340TCMRPNATEGS  IDDG OHAGNV IAOGHSRETQTTYPCEI      C12591
C         HVPRRTC D    D 00004  DDCPR  LP  DCLT TDRG R MAST    E2   EE M     C12592
C                                                                            012593
C                                                                            C12594
C                                                                            C12595
C                                                                            012596
C*************************************************************                012597
C                                                                            012598
C     SUBROUTINE ADOT                DEBUG # 105                             012599
C                                             USER DEFINED SUBROUTINE        012600
C        CALLS:                                                              012601
C                   (NONE UNLESS USER REQUESTED)                             C12602
C        CALLED BY:                                                          012603
C                   YDOT                                                     C12604
C        PURPOSE:                                                            C12605
C                   TO DEFINE THE ACCELERATION OF COORDINATE IC WHICH IS     C12606
C                   RHEONOMICALLY CONSTRAINED, THAT IS TO PROVIDE THE        C12607
C                   ALPHA DOUBLE DOT TERM IN EQUATION II-6, VOL I A          C12608
C                   USER DEFINED ROUTINE                                     C12609
C                                                                            C12610
C*************************************************************                C12611
C                                                                            012612
C     SUBROUTINE ADD3               DEBUG # 27                               C12613
C                                             UTILITY, MATRIX OPERATION      012614
C        CALLS:                                                              012615
C                   (NONE)                                                   C12616
C        CALLED BY:                                                          012617
C                   CREL                                                     C12618
C                   CREC                                                     C12619
C        PURPOSE:                                                            C12620
C                   MULTIPLY THREE MATRICES EACH BY A SCALAR AND ADD.        012621
C                   FORM THE SUM:                                            012622
C                         AZ = SA*A + SB*B + SC*C                            C12623
```

```
C                        WHERE
C                          A,B,C     ARE INPUT (NR,NC) MATRICES
C                          SA,SB,SC ARE INPUT SCALARS
C                          B,C       ARE NOT DESTROYED, INPUT A IS
C                          AZ        OUTPUT MATRIX RETURNED IN A ARRAY
C
C**********************************************************
C
C      SUBROUTINE ADT                  DEBUG # 104
C                                                USER DEFINED SUBROUTINE
C         CALLS:
C                        (NONE UNLESS USER REQUESTED)
C         CALLED BY:
C                        FINDD
C         PURPOSE:
C                        TO DEFINE THE VELOCITY OF COORDINATE IC WHICH IS
C                        RHENOMICALLY CONSTRAINED, THAT IS TO PROVIDE THE
C                        ALPHA DOT TERM IN EQUATION II-5 VOL 1. A USER
C                        DEFINED ROUTINE
C                        CAUTION TO AVOID EXTRANEOUS IMPULSE EFFECTS
C                        INTEGRATE ALPHA DOUBLE DOT TO GET ALPHA DOT
C                        PROBLEMS CAN ARISE DUE TO NUMERICAL INTEGRATION
C
C**********************************************************
C
C      SUBROUTINE ALPHAA               DEBUG # 95
C                                                UTILITY, MATRIX OPERATION
C         CALLS:
C                        (NONE)
C         CALLED BY:
C                        (NONE)
C         PURPOSE:
C                        MULTIPLY SCALAR SA BY MATRIX A, RETURN Z
C                                  Z = SA*A
C                        A IS NOT DESTROYED
C
C**********************************************************
C
C      SUBROUTINE ASIMLR               DEBUG # 65
C                                                MAIN LINE OPERATION
C         CALLS:
C                        ZERO       WRITIS     WRITES
C                        REVADD     FINDT      WRITE
C                        MULTA      GAUSSI     MULTB
C         CALLED BY:
C                        DYN340
C         PURPOSE:
C                        ESTABLISH TRANSFORMED PARTIAL DERIVATIVE MATRIX
C                        BY PERFORMING A SIMILARITY TRANSFORMATION TO
C                        EXCHANGE PLANT STATE VARIABLES Y FOR SENSOR SIGNALS
C                        XSS AND CONTROL SYSTEM VARIABLES DELTA FOR TORQUE
C                        SIGNALS U. SEE SECTION 'EXCHANGE OF VARIABLES '
C                        PAGE 59 VOL I FOR BACKGROUND THEORY
C
C**********************************************************
C
C      SUBROUTINE ASGR                 DEBUG # 118
C                                                UTILITY, MATRIX OPERATION
C         CALLS:
C                        (NONE)
```

```
C          CALLED BY:                                                   012684
C                    [illegible]                                        012685
C          PURPOSE:                                                     012686
C                    COMPUTE SQUARE OF THE MATRIX A, RETURNS A**2 IN B  012687
C                              B = A*A                                  012688
C                    A IS NOT DESTROYED                                 012689
C                                                                       012690
C*****************************************************************       012691
C                                                                       012692
C       SUBROUTINE [illegible]           DEFCG # 56                     012693
C                                       UTILITY, MATRIX OPERATION       012694
C          CALLS:                                                       012695
C                    (NONE)                                             012696
C          CALLED BY :                                                  012697
C                    (NONE)                                             012698
C          PURPOSE:                                                     012699
C                    FORM THE SPECIAL MATRIX TRIPLE PRODUCT             012700
C                              Z = B*X*B**T                             012701
C                    X MATRIX (NCV,NCV) SYMMETRIC                       012702
C                    B MATRIX (NRB,NCV)                                 012703
C                    Z MATRIX (NRB,NRB) WILL BE SYMMETRIC               012704
C                    NOTHING DESTROYED                                  012705
C                                                                       012706
C*****************************************************************       012707
C                                                                       012708
C       SUBROUTINE [illegible]           DEFCG # 52                     012709
C                                       UTILITY, MATRIX OPERATION       012710
C          CALLS:                                                       012711
C                    WRITES                                             012712
C          CALLED BY:                                                   012713
C                    YOUT                                               012714
C          PURPOSE:                                                     012715
C                    SPECIAL OPERATION TO USE DECOMPOSED GMG AND D      012716
C                    MATRICES OBTAINED VIA DCOM2 ALONG WITH THE V MATRIX 012717
C                    COMPUTED IN YOUT TO COMPUTE LAGRANGE MULTIPLIERS   012718
C                                                                       012719
C*****************************************************************       012720
C                                                                       012721
C       SUBROUTINE BOUTQP                DEFCG # 51                     012722
C                                       MAIN LINE OPERATION             012723
C          CALLS:                                                       012724
C                    WRITES    SKEWVS                                   012725
C                    MULTAD                                             012726
C                    MULTS                                              012727
C          CALLED BY:                                                   012728
C                    YOUT                                               012729
C          PURPOSE:                                                     012730
C                    GET THE LOWER CASE E DOT MATRIX IN EQUATION II-6,  012731
C                    REFER TO APPENDIX C VOL I FOR SUPPORTING THEORY    012732
C                    ALSO PAGES 36-38, VOL I                            012733
C                                                                       012734
C*****************************************************************       012735
C                                                                       012736
C       SUBROUTINE BBGENR                DEFCG # 45                     012737
C                                       MAIN LINE OPERATION             012738
C          CALLS:                                                       012739
C                    MULTS                                              012740
C                    SKEWVS                                             012741
C                    WRITES                                             012742
C          CALLED BY:                                                   012743
```

```
C                   DYNSE0                                                012744
C                   YDOT                                                  012745
C        PURPOSE:                                                         012746
C                   USED TO COMPUTE THE ELEMENTS OF THE BH ARRAY THIS     012747
C                   ARRAY DEFINES HINGE POINT KINEMATICS AND IS DEFINED   012748
C                   IN VOL I BY EQUATIONS II-82 AND II-83                 012749
C                                                                         012750
C                                                                         012751
C*********************************************************************    012752
C                                                                         012753
C     SUBROUTINE BSGENR            DEBUG # 47                             012754
C                                                MAIN LINE OPERATION      012755
C        CALLS:                                                           012756
C                   MULT3                                                 012757
C                   SKEWV3                                                012758
C        CALLED BY:                                                       012759
C                   YDOT                                                  012760
C        PURPOSE:                                                         012761
C                   COMPUTE THE ELEMENTS OF THE BS ARRAY THIS DEFINES     012762
C                   THE KINEMATIC COEFFICIENTS FOR ALL SENSOR POINT       012763
C                   REFERENCE FRAMES, SEE PAGE 17 VOL II                  012764
C                                                                         012765
C*********************************************************************    012766
C                                                                         012767
C     SUBROUTINE BTABA             DEBUG # 35                             012768
C                                                UTILITY, MATRIX OPERATION 012769
C        CALLS:                                                           012770
C                   (NONE)                                                012771
C        CALLED BY:                                                       012772
C                   MSMODE                                                012773
C        PURPOSE:                                                         012774
C                   FORM THE PARTICULAR MATRIX TRIPLE PRODUCT             012775
C                        AZ = B**T * A * B                                012776
C                   INPUT MATRICES A AND B, OUTPUT AZ, MATRIX A           012777
C                   IS DESTROYED                                          012778
C                                                                         012779
C*********************************************************************    012780
C                                                                         012781
C     SUBROUTINE CANCLR            DEBUG # 78                             012782
C                                                UTILITY, NON-MATRIX OPER. 012783
C        CALLS:                                                           012784
C                   (NONE)                                                012785
C        CALLED BY:                                                       012786
C                   DYNS40                                                012787
C        PURPOSE:                                                         012788
C                   TO CANCEL OUT OF THE ROOTS ARRAY R THE ROOTS OF THE   012789
C                   TRANSFER FUNCTION WHICH ARE COMMON TO BOTH            012790
C                   NUMERATOR AND DENOMINATOR                             012791
C                                                                         012792
C*********************************************************************    012793
C                                                                         012794
C     SUBROUTINE COMENT            DEBUG # 7                              012795
C                                                UTILITY, DATA CONTROL, I/O 012796
C        CALLS:                                                           012797
C                   PAGEHD                                                012798
C        CALLED BY:                                                       012799
C                   MAIN                                                  012800
C        PURPOSE:                                                         012801
C                   TO READ IN AND OUTPUT ON THE FIRST PAGE OF THE        012802
C                   OUTPUT DATA ANY NUMBER OF USER WRITTEN COMMENT CARDS  012803
```

```
C                                                                        012804
C*****************************************************************       012805
C                                                                        012806
C     SUBROUTINE CONTRL            DEBUG # 107                           012807
C                                             USER DEFINED SUBROUTINE    012808
C        CALLS:                                                          012809
C                   TFPLY     ZERO                                       012810
C                   READ      (OTHERS WHICH ARE USER REQUESTED)          012811
C                   WRITE                                                012812
C        CALLED BY:                                                      012813
C                   TORQUE                                               012814
C        PURPOSE:                                                        012815
C                   1)ACCEPT TRANSFER FUNCTION DESCRIPTION OF LINEAR     012816
C                     CONTROL SYSTEM                                     012817
C                   2)DEFINE ALL AUXILIARY DIFFERENTIAL EQUATIONS        012818
C                     REQUIRED FOR SIMULATION PURPOSES (USER DEFINED)    012819
C                                                                        012820
C*****************************************************************       012821
C                                                                        012822
C     SUBROUTINE CRLA              DEBUG # 21                            012823
C                                             UTILITY, DATA PROCESSING   012824
C        CALLS:                                                          012825
C                   ZERO      WRITE                                      012826
C                   SATO                                                 012827
C                   FETCH                                                012828
C        CALLED BY:                                                      012829
C                   MSMODL                                               012830
C        PURPOSE:                                                        012831
C                   MAKES USE OF LUMPED PARAMETER DATA TO COMPUTE THE    012832
C                   ALPHA COEFFICIENTS IN EQ II-55 WHICH ARE ASSOCIATED  012833
C                   WITH THE COMPUTATION OF THE CHANGE IN STATIC MASS    012834
C                   MOMENT DUE TO FLEXIBLE BODY DEFORMATION, SEE ALSO    012835
C                   APPENDIX-A VOL I EQS. A-12 TO A-20                   012836
C                                                                        012837
C                                                                        012838
C*****************************************************************       012839
C                                                                        012840
C     SUBROUTINE CRADJ             DEBUG # 19                            012841
C                                             UTILITY, DATA PROCESSING   012842
C        CALLS:                                                          012843
C                   WRITE                                                012844
C                   FETCH                                                012845
C                   SATO                                                 012846
C        CALLED BY:                                                      012847
C                   MSMODL                                               012848
C        PURPOSE:                                                        012849
C                   MAKES USE OF LUMPED PARAMETER DATA TO COMPUTE THE    012850
C                   D AND A COEFFICIENTS WHICH ARE INDEPENDENT OF        012851
C                   DEFORMATION AND DEFINE THE INERTIAL COUPLING BETWEEN 012852
C                   D ROTATION AND TRANSLATION COORDINATES AND THE       012853
C                   IRSPEX(I) MODAL DISPLACEMENT COORDINATES.            012854
C                   SEE EQ II-67                                         012855
C                                                                        012856
C                                                                        012857
C*****************************************************************       012858
C                                                                        012859
C     SUBROUTINE CRMB              DEBUG # 22                            012860
C                                             UTILITY, DATA PROCESSING   012861
C        CALLS:                                                          012862
C                   ZERO      SATO                                       012863
```

```
C                  WRITE                                               012864
C                  FETCH                                               012865
C       CALLED BY                                                      012866
C                  MSMODL                                              012867
C        PURPOSE:                                                      012868
C                  MAKES USE OF LUMPED PARAMETER DATA TO COMPUTE THE    012869
C                  B COEFFICIENTS IN EQ II-88 WHICH ARE ASSOCIATED      012870
C                  WITH THE COMPUTATION OF THE CHANGE IN THE BODY'S     012871
C                  INERTIA TENSOR DUE TO TERMS LINEAR IN FLEXIBLE BODY  012872
C                  DEFORMATION, SEE APPENDIX A VOL 1 EQS A-21 TO A-26   012873
C                                                                       012874
C                                                                       012875
C**********************************************************             012876
C                                                                       012877
C     SUBROUTINE CREC             DEBUG # 23                            012878
C                                           UTILITY, DATA PROCESSING    012879
C        CALLS:                                                         012880
C                  FETCH     ZERO                                       012881
C                  DATB      WRITE                                      012882
C                  ADD3                                                 012883
C        CALLED BY:                                                     012884
C                  MSMODL                                               012885
C        PURPOSE:                                                       012886
C                  MAKES USE OF LUMPED PARAMETER DATA TO COMPUTE THE    012887
C                  C COEFFICIENTS IN EQ II-88 WHICH ARE ASSOCIATED      012888
C                  WITH THE COMPUTATION OF THE CHANGE IN THE COUPLING   012889
C                  COEFFICIENTS BETWEEN ROTATION AND DISPLACEMENT       012890
C                  COORDINATES DUE TO TERMS LINEAR IN FLEXIBLE BODY     012891
C                  DEFORMATION, SEE APPENDIX A VOL 1 EQS A-27,28,29     012892
C                  ALSO COMPUTES THE C COEFFICIENTS IN EQ II-89 WHICH   012893
C                  ARE ASSOCIATED WITH THE COMPUTATION OF THE CHANGE    012894
C                  IN THE BODY'S INERTIA TENSOR DUE TO TERMS QUADRATIC  012895
C                  IN FLEXIBLE BODY DEFORMATION, SEE APPENDIX A VOL 1   012896
C                  EQS A-30 TO A-35                                     012897
C                                                                       012898
C**********************************************************             012899
C                                                                       012900
C     SUBROUTINE CREE             DEBUG # 20                            012901
C                                           UTILITY, DATA PROCESSING    012902
C        CALLS:                                                         012903
C                  WRITE     ADD3                                       012904
C                  FETCH                                                012905
C                  DATB                                                 012906
C        CALLED BY:                                                     012907
C                  MSMODL                                               012908
C        PURPOSE:                                                       012909
C                  MAKES USE OF LUMPED PARAMETER DATA TO COMPUTE THE    012910
C                  E COEFFICIENTS WHICH ARE INDEPENDENT OF DEFORMATION  012911
C                  THESE DEFINE THE MODAL MASS MATRIX WHICH IS DIAGONAL 012912
C                  IF THE MODES ARE ORTHOGONAL SEE EQ II-87 MODES MAY   012913
C                  BE NON-ORTHOGONAL                                    012914
C                                                                       012915
C                                                                       012916
C**********************************************************             012917
C                                                                       012918
C     SUBROUTINE CREMOD           DEBUG # 18                            012919
C                                           UTILITY, DATA PROCESSING    012920
C        CALLS:                                                         012921
C                  FETCH                                                012922
C                  DATB                                                 012923
```

```
C                  WRITE                                               012924
C         CALLED BY:                                                   012925
C                  MSMODL                                              012926
C         PURPOSE:                                                     012927
C                  MAKES USE OF LUMPED PARAMETER DATA TO COMPUTE THE   012928
C                  FLEXIBLE BODY'S UNDEFORMED INERTIA TENSOR, STATIC   012929
C                  MASS MOMENT AND TOTAL MASS. SEE EQ II-67            012930
C                                                                      012931
C                                                                      012932
C*****************************************************************     012933
C                                                                      012934
C      SUBROUTINE CRIT3               DEBUG # 17                       012935
C                                        UTILITY, DATA PROCESSING      012936
C         CALLS:                                                       012937
C                  FETCH                                               012938
C                  STORE                                               012939
C         CALLED BY:                                                   012940
C                  MSMODL                                              012941
C         PURPOSE:                                                     012942
C                  TO FETCH RAW FLEXIBLE BODY DATA FROM NTAPE2 AND TO  012943
C                  CREATE THE FUNCTIONS WHICH MUST BE INTEGRATED OVER  012944
C                  THE BODY'S VOLUMN TO DEFINE RESULTANT MODE DEPENDENT 012945
C                  INERTIA PARAMETERS                                  012946
C                                                                      012947
C*****************************************************************     012948
C                                                                      012949
C      SUBROUTINE DCOM2               DEBUG # 42                       012950
C                                        UTILITY, MATRIX OPERATION     012951
C         CALLS:                                                       012952
C                  (NONE)                                              012953
C         CALLED BY:                                                   012954
C                  YOST                                                012955
C         PURPOSE:                                                     012956
C                  DECOMPOSES A TO FORM AN UPPER TRIANGULAR MATRIX     012957
C                  WITH ONES ON DIAGONAL AND ZEROES BELOW SUCH THAT    012958
C                           A = U**T * D * U                           012959
C                  A REAL, SQUARE, AND SYMMETRIC. ROUTINE ACTUALLY     012960
C                  LEAVES GARBAGE ON DIAGONAL AND BELOW                012961
C                  USED IN CONJUNCTION WITH BAKSLV TO OBTAIN           012962
C                  LAGRANGE MULTIPLIERS                                012963
C                                                                      012964
C*****************************************************************     012965
C                                                                      012966
C      SUBROUTINE DCGRRT              DEBUG # 70                       012967
C                                        UTILITY, NON-MATRIX OPER.     012968
C         CALLS:                                                       012969
C                  (NONE)                                              012970
C         CALLED BY:                                                   012971
C                  DYN340                                              012972
C         PURPOSE:                                                     012973
C                  TAKES COMPLEX ROOTS (RR + I*RI) FOUND IN QR ROUTINES 012974
C                  PLACES REAL AND ZERO ROOTS IN RCRT ARRAY,           012975
C                  COUNTS THE NUMBER OF ZEROES (KZ), AND PLACES        012976
C                  COMPLEX PAIRS IN CMPR ARRAY. (KC=NUMBER OF PAIRS)   012977
C                                                                      012978
C*****************************************************************     012979
C                                                                      012980
C      SUBROUTINE DEFI                DEBUG # 112                      012981
C                                        DEFINITIONS                   012982
C         CALLS:                                                       012983
```

```
C                   (NONE)
C        CALLED BY :
C                   (NONE)
C        PURPOSE:
C                   DEFINE COMMON BLOCKS AND PARAMETERS STORED THEREIN
C
C***********************************************************
C
C     SUBROUTINE DEF2              DEBUG # 113
C                                              DEFINITIONS
C        CALLS:
C                   (NONE)
C        CALLED BY :
C                   (NONE)
C        PURPOSE:
C                   DEFINE ALL SUBROUTINES AS TO PURPOSE
C
C***********************************************************
C
C     SUBROUTINE DEF3              DEBUG # 114
C                                              DEFINITIONS
C        CALLS:
C                   (NONE)
C        CALLED BY :
C                   (NONE)
C        PURPOSE:
C                   DEFINE REQUIRED INPUT DATA SET UP AND FORMAT
C
C***********************************************************
C
C     SUBROUTINE DEF4              DEBUG # 115
C                                              DEFINITIONS
C        CALLS:
C                   (NONE)
C        CALLED BY :
C                   (NONE)
C        PURPOSE:
C                   DEFINE OUTPUT DATA
C
C***********************************************************
C
C     SUBROUTINE DEF5              DEBUG # 116
C                                           DEFINITIONS
C        CALLS:
C                  (NONE)
C        CALLED BY:
C                  (NONE)
C        PURPOSE:
C                  DEFINE ALL PARAMETERS COMPUTED IN DISCUS
C
C***********************************************************
C
C     SUBROUTINE DFORMB            DEBUG # 80
C                                              UTILITY, NON-MATRIX OPER.
C        CALLS:
C                   (NONE)
C        CALLED BY:
C                   DYNS40
C        PURPOSE:
C                   TAKES REAL ROOTS DEFINED IN ARRAY RERT AND COMPUTES
```

```
C                 ASSOCIATED TIME CONSTANTS FOR = -1.0/HERT                  013044
C                 TAKES COMPLEX PAIRS DEFINED IN ARRAY CMPR, COMPUTES        013045
C                       FDC(L) = OMEGA                                       013046
C                       FDC(L-1) = ZETA                                      013047
C                 WHERE COMPLEX PAIR ARE ROOTS OF:                           013048
C                       S**2 + 2*ZETA*OMEGA*S + OMEGA**2 = 0                 013049
C                 ALSO FORMS PRODUCT OF ALL ROOTS FOR BODE GAIN              013050
C                 COMPUTATION                                                013051
C                                                                            013052
C*****************************************************************           013053
C                                                                            013054
C     FUNCTION DIMAG(S)             DEBUG # 76                               013055
C                                            UTILITY, NON-MATRIX OPER.       013056
C        CALLS:                                                              013057
C                 (NONE)                                                     013058
C        CALLED BY:                                                          013059
C                 REDCUS                                                     013060
C                 SFREQ2                                                     013061
C        PURPOSE:                                                            013062
C                DIMAG(S) - IMAGINARY PART OF COMPLEX NUMBER S               013063
C                           PRECISION REAL*8                                 013064
C                                                                            013065
C*****************************************************************           013066
C                                                                            013067
C     FUNCTION DREAL(S)             DEBUG # 77                               013068
C                                            UTILITY, NON-MATRIX OPER.       013069
C        CALLS:                                                              013070
C                 (NONE)                                                     013071
C        CALLED BY:                                                          013072
C                 REDCUS                                                     013073
C                 SFREQ2                                                     013074
C        PURPOSE:                                                            013075
C                 DREAL(S) - REAL PART OF COMPLEX NUMBER S                   013076
C                            PRECISION REAL*8                                013077
C                                                                            013078
C*****************************************************************           013079
C                                                                            013080
C     SUBROUTINE DYNSID             DEBUG # 2                                013081
C                                            MAIN LINE EXECUTIVE             013082
C        CALLS:                                                              013083
C                 RSMODE    RSMODL    PASCHD                                 013084
C                 READIN    READ      WRITES                                 013085
C                 WRITIS    RRIGID                                           013086
C        CALLED BY:                                                          013087
C                 MAIN                                                       013088
C        PURPOSE:                                                            013089
C                 TO BUILD THE SIMULATION MODEL FROM THE INPUT DATA          013090
C                 SEE FLOW CHART PAGE 23 VOL II. THE FOLLOWING               013091
C                 INPUT DATA IS READ:                                        013092
C                  1) BASIC MODEL SIZES AND TOPOLOGY                         013093
C                  2) BODY TYPES, NUMBER OF MODES                            013094
C                  3) SENSOR POINT LOCATION                                  013095
C                  4) CONSTRAINT CONDITIONS                                  013096
C                  5) CONTINUOUS HINGE FRAME DATA, ORIENTATION + RATE        013097
C                  6) REACTION WHEEL DATA                                    013098
C                  7) TIME HISTORY DATA                                      013099
C                  8) ORBIT, [illegible] CONTROL                             013100
C                  9) CONTROL SYSTEM DATA                                    013101
C                 10) GRAVITY GRADIENT DATA                                  013102
C                 11) CALL RRIGID TO READ RIGID BODY CHARACTERISTICS         013103
```

```
C                    12) CALL MSMODE TO READ FLEXIBLE BODY LUMPED              013104
C                        PARAMETER DATA                                        013105
C                    13) CALL MSMODE TO READ FLEXIBLE BODY CONSISTANT          013106
C                        MASS PARAMETER DATA                                   013107
C                    WITH THIS DATA SOME DATA SET ARRAYS ARE COMPUTED IN       013108
C                    DYNS10. AN ECHO OF ALL INPUT DATA AND COMPUTED DATA       013109
C                    IS GIVEN. SEE SUBROUTINES DEF3 AND DEF4 LISTED IN         013110
C                    VOL 2                                                     013111
C                                                                              013112
C*********************************************************************         013113
C                                                                              013114
C     SUBROUTINE DYNS20            DEBUG # 37                                  013115
C                                              MAIN LINE EXECUTIVE             013116
C        CALLS:                                                                013117
C                    RKADAM     YDOT       ROTDM                               013118
C                    FINDU      DMGENR     PRNTOU                              013119
C                    LINEAR     ENGMOM                                         013120
C                    PLOTWR     WRITIS                                         013121
C        CALLED BY:                                                            013122
C                    MAIN                                                      013123
C        PURPOSE:                                                              013124
C                 1) READ NTAPE1 ON WHICH ALL TIME INDEPENDENT RIGID AND       013125
C                     FLEXIBLE BODY DATA HAS BEEN WRITTEN.                     013126
C                 2) LOAD INITIAL CONDITIONS INTO STATE VECTOR.                013127
C                 3) COMPUTE INITIAL KINEMATIC DATA (DMGENR,ROTDM)             013128
C                 4) FIND INDEPENDENT COORDINATES (FINDU)                      013129
C                 5) GET TIME DERIVATIVE OF STATE VECTOR (YDOT)                013130
C                 6) COMPUTE ENERGY AND MOMENTUM FOR BODIES AND TOTAL          013131
C                     SYSTEM (ENGMOM)                                          013132
C                 7) INTEGRATE NON-LINEAR EQUATIONS OF MOTION (RKADAM)         013133
C                 8) LINEARIZE COUPLED EQUATIONS OF MOTION (LINEAR)            013134
C                 9) OUTPUT DATA ON PRINTER OR PLOTTER (PRNTOU,PLOTWR)         013135
C                                                                              013136
C*********************************************************************         013137
C                                                                              013138
C     SUBROUTINE DYNS30            DEBUG # 62                                  013139
C                                              MAIN LINE EXECUTIVE             013140
C        CALLS:                                                                013141
C                    PLTCAN                                                    013142
C                    PAGEHD                                                    013143
C        CALLED BY:                                                            013144
C                    MAIN                                                      013145
C        PURPOSE:                                                              013146
C                    READ DATA REQUIRED TO SETUP PLOT TAPE FOR SC4020          013147
C                                                                              013148
C*********************************************************************         013149
C                                                                              013150
C     SUBROUTINE DYNS40            DEBUG # 64                                  013151
C                                              MAIN LINE EXECUTIVE             013152
C        CALLS:                                                                013153
C                    ASIMEX     NOMS       READIM     RAWITE     TTFF          013154
C                    CANCEL     NYPLOT     RLOCUS     SFREQ2     WRITE         013155
C                    DCGAIN     PAGEHD     REPLOT     SIFT       ZERO          013156
C                    CTRESP     GRDRVR     RMVZRO     SPLOT      DFORMB        013157
C                    NIPLOT     READ       RTEP       T-TYPE                   013158
C        CALLED BY:                                                            013159
C                    MAIN                                                      013160
C        PURPOSE:                                                              013161
C                    ALL ANALYSIS ASSOCIATED WITH THE LINEARIZED               013162
C                     EQUATIONS DIRECTED BY DYNS40                             013163
```

```
C                  1) READ LINEARIZED PARTIAL DERIVATIVE MATRIX FROM        013164
C                     [illegible]                                           013165
C                  2) FIND EIGENVALUES OF LINEARIZED SYSTEM (RORVR)         013166
C                  3) PERFORM SIMILARITY TRANSFORMATION TO EXCHANGE         013167
C                     [illegible] STATE VARIABLES FOR [illegible] AND TORQUE 013168
C                     [illegible]                                           013169
C                  4) FIND EIGENVECTORS OF TRANSFORMED EQUATIONS,           013170
C                     [illegible] SAME AS ABOVE (RORVR)                     013171
C                  5) READ DATA CARD TO DEFINE TYPE OF ANALYSIS             013172
C                  6) CALC FORCED [illegible] TIME HISTORY RESPONSE (GTRESP) 013173
C                  7) READ CARDS TO DEFINE TYPE OF TRANSFER FUNCTION,       013174
C                     SELECT INPUT VARIABLES, OUTPUT DISPLAY                013175
C                  8) OBTAIN CHARACTERISTIC MATRIX AND CORRESPONDING        013176
C                     INPUT COEFFICIENTS FOR TRANSFER FUNCTION (TFTYPE)     013177
C                  9) OBTAIN ROOTS OF DENOMINATOR (GRORVR)                  013178
C                 10) OBTAIN ROOTS OF NUMERATOR (NUMRS)                     013179
C                 11) SET UP ROOTS ARRAY (LIFT, [illegible], [illegible], RNVZRC, 013180
C                     AND ITFF)                                             013181
C                 12) CANCEL OUT ROOTS COMMON TO NUMERATOR AND              013182
C                     DENOMINATOR (CANCEL)                                  013183
C                 13) COMPUTE FREQUENCY RESPONSE, DISPLAY DATA VIA          013184
C                     BODE, NICHOLS, AND/OR NYQUIST (BPLOT, NIPLOT, NYPLOT) 013185
C                 14) CONVERT ROOTS TO POLYNOMIAL (RTOP)                    013186
C                 15) PERFORM ROOT LOCUS ANALYSIS (ROLOCUS, RLPLOT)         013187
C                 16) COMPUTE EIGENVECTORS OF CHARACTERISTIC MATRIX         013188
C                     FOR SELECTED EIGENVALUES                              013189
C                                                                           013190
C****************************************************************           013191
C                                                                           013192
C       SUBROUTINE EIGVEC            DREQ # 117                             013193
C                                         UTILITY, MATRIX OPERATION         013194
C          CALLS:                                                           013195
C                  (NONE)                                                   013196
C          CALLED BY:                                                       013197
C                  DYN540                                                   013198
C          PURPOSE:                                                         013199
C                  TO FIND THE EIGENVECTORS OF A NON-SYMMETRIC MATRIX       013200
C                  BY A MODIFIED WILKINSON'S INVERSE ITERATION METHOD       013201
C                  SUBROUTINE ACCEPTS MATRIX A, A**2 AND EIGENVALUE         013202
C                  LAMBDA FROM WHICH IT COMPUTES EIGENVECTOR X OF           013203
C                                 A*X = LAMBDA*X                            013204
C                  AND THE TRANSPOSED EIGENVECTOR V OF                      013205
C                                 A**T*V = LAMBDA*V                         013206
C                  EIGENVECTORS ASSOCIATED WITH REPEATED EIGENVALUES        013207
C                  CANNOT BE FOUND                                          013208
C                                                                           013209
C****************************************************************           013210
C                                                                           013211
C       SUBROUTINE ENGMOM            DREQ # 56                              013212
C                                         MAIN LINE OPERATION               013213
C          CALLS:                                                           013214
C                  MULTD                                                    013215
C                  MULTAD                                                   013216
C          CALLED BY:                                                       013217
C                  DYN520                                                   013218
C          PURPOSE:                                                         013219
C                  TO COMPUTE LINEAR AND ANGULAR MOMENTUM ALONG WITH        013220
C                  KINETIC AND POTENTIAL ENERGY FOR EACH BODY AND           013221
C                  FOR THE TOTAL SYSTEM                                     013222
C                                                                           013223
```

```
C*************************************************************         013224
C                                                                      013225
C      SUBROUTINE [illegible]          DEBUG # 110                     013226
C                                       USER DEFINED SUBROUTINE        013227
C         CALLS:                                                       013228
C                    (NONE UNLESS USER REQUESTED)                      013229
C         CALLED BY:                                                   013230
C                    TORQUE                                            013231
C         PURPOSE:                                                     013232
C                    TO DEFINE THE AUXILIARY EQUATIONS WHICH WILL DEFINE   013233
C                    THE SENSOR SIGNAL AND TORQUE SIGNAL EQUATIONS TO BE   013234
C                    USED FOR STABILITY ANALYSIS (USER DEFINED)        013235
C                                                                      013236
C*************************************************************         013237
C                                                                      013238
C      SUBROUTINE EXTOR            DEBUG # 108                         013239
C                                       USER DEFINED SUBROUTINE        013240
C         CALLS:                                                       013241
C                    (NONE UNLESS USER REQUESTED)                      013242
C         CALLED BY:                                                   013243
C                    TORQUE                                            013244
C         PURPOSE:                                                     013245
C                    TO DEFINE ALL EXTERNAL FORCE/TORQUE LOADS.        013246
C                    THESE ACT AT SENSOR POINTS                        013247
C                                                                      013248
C*************************************************************         013249
C                                                                      013250
C      SUBROUTINE FETCH            DEBUG # 25                          013251
C                                       UTILITY, DATA CONTROL, I/O     013252
C         CALLS:                                                       013253
C                    (NONE)                                            013254
C         CALLED BY:                                                   013255
C                    TRACOL    CREE                                    013256
C                    CRETJ     CREA                                    013257
C                    CRPMCD    CREB                                    013258
C                    CRPADD    CREC                                    013259
C         PURPOSE:                                                     013260
C                    TO FETCH A PARTICULAR MATRIX FROM TAPE. THE TAPE IS   013261
C                    ASSUMED TO BE WRITTEN IN BINARY AND THE RECORD    013262
C                    NUMBER ASSOCIATED WITH THE MATRIX KNOWN           013263
C                                                                      013264
C*************************************************************         013265
C                                                                      013266
C      SUBROUTINE FINDT            DEBUG # 66                          013267
C                                       MAIN LINE OPERATION            013268
C         CALLS:                                                       013269
C                    ZERO      WRITIS                                  013270
C                    PAGEHD    WRITES                                  013271
C                    REVADD                                            013272
C         CALLED BY:                                                   013273
C                    ASIMER                                            013274
C         PURPOSE:                                                     013275
C                    COMPUTE THE SIMILARITY TRANSFORMATION MATRIX R OF 013276
C                    EQUATION III-23. VOL I. SEE PAGE 63 VOL I ALSO    013277
C                                                                      013278
C                                                                      013279
C*************************************************************         013280
C      SUBROUTINE FINDD            DEBUG # 38                          013281
C                                       MAIN LINE OPERATION            013282
C         CALLS:                                                       013283
```

```
C                   ADT                                                  013284
C                   WRITIS                                               013285
C                   WRITES                                               013286
C        CALLED BY:                                                      013287
C                   DYNS20                                               013288
C                   YDOT                                                 013289
C        PURPOSE:                                                        013290
C                   IN DYNS20 CALL FINDU TO COMPUTE THE INDEPENDENT      013291
C                   COMPONENTS OF THE U VECTORS AT T=0 FOR INSERTION     013292
C                   INTO THE STATE VECTOR.                               013293
C                   IN YDOT CALL FINDU TO COMPUTE THE DEPENDENT          013294
C                   COORDINATES AT EACH TIME STEP FOR INSERTION INTO     013295
C                   THE STATE VECTOR                                     013296
C                   ONLY INDEPENDENT COORDINATES ARE INTEGRATED          013297
C                   DEPENDENT COORDINATES MUST BE COMPUTED IN FINDU      013298
C                                                                        013299
C************************************************************            013300
C                                                                        013301
C     SUBROUTINE FIT                DEBUG # 75                           013302
C                                              UTILITY, NON-MATRIX OPER. 013303
C        CALLS:                                                          013304
C                   (NONE)                                               013305
C        CALLED BY:                                                      013306
C                   REDUCS                                               013307
C        PURPOSE:                                                        013308
C                   FIND THE SLOPE OF THE LEAST SQUARES FIT LINE THROUGH 013309
C                   THE LOG OF THE COEFFICIENTS OF A POLYNOMIAL. USED    013310
C                   FOR SCALING TO RETAIN NUMERICAL PRECISION            013311
C                                                                        013312
C************************************************************            013313
C                                                                        013314
C     SUBROUTINE FORMB              DEBUG # 79                           013315
C        CALLS:                                                          013316
C                   (NONE)                                               013317
C        CALLED BY:                                                      013318
C                   (NONE)                                               013319
C        PURPOSE:                                                        013320
C                   NEARLY THE SAME AS DFORMB BUT WITHOUT GAIN           013321
C                   COMPUTATION                                          013322
C                                                                        013323
C************************************************************            013324
C                                                                        013325
C     SUBROUTINE GAUSSI             DEBUG # 11                           013326
C                                              UTILITY, MATRIX OPERATION 013327
C        CALLS:                                                          013328
C                   WRITES                                               013329
C        CALLED BY:                                                      013330
C                   ASRQSC    EIRESP                                     013331
C                   SSIMCR                                               013332
C                   [illegible]                                          013333
C        PURPOSE:                                                        013334
C                   MATRIX INVERSION SUBROUTINE, RETURNS MATRIX INVERSE  013335
C                   ALONG WITH PIVOT ELEMENTS USED TO OBTAIN IT. THESE   013336
C                   MAY BE USED TO OBTAIN THE DETERMINANT OF THE MATRIX  013337
C                                                                        013338
C************************************************************            013339
C                                                                        013340
C     SUBROUTINE SETSMG             DEBUG # 49                           013341
C                                              MAIN LINE OPERATION       013342
C        CALLS:                                                          013343
```

```
C                   WRITES
C                   MLTSR
C       CALLED BY:
C                   YDOT
C       PURPOSE:
C                   COMPUTE
C                         GMB = B*M**(-1)*B**T
C                   OF EQUATION II-6 VOL I. SAVE ALSO
C                         BM  = B*M**(-1)
C                   BOTH USED IN LAGRANGE MULTIPLIER COMPUTATION
C
C***********************************************************
C
C     SUBROUTINE GMISC               DEBUG # 111
C                                             USER DEFINED SUBROUTINE
C       CALLS:
C                   (NONE UNLESS USER REQUESTED)
C       CALLED BY:
C                   TORQUE
C       PURPOSE:
C                   A CATCH-ALL SUBROUTINE (USER DEFINED) USED TO ADD
C                   TO THE R.H.S. OF THE EQUATIONS OF MOTION ANY
C                   ADDITIONAL CONTRIBUTIONS SUCH AS THEMALLY INDUCED
C                   DEFORMATION EFFECTS
C
C***********************************************************
C
C     SUBROUTINE GRVGRD              DEBUG # 50
C                                             MAIN LINE OPERATION
C       CALLS:
C                   WRITES
C                   MULTS
C                   MULTAD
C       CALLED BY:
C                   YDOT
C       PURPOSE:
C                   COMPUTE THE CONTRIBUTION TO THE FORCE VECTOR FROM
C                   GRAVITY GRADIENT EFFECTS
C
C***********************************************************
C
C     SUBROUTINE INTAPE              DEBUG # 9
C                                             UTILITY, DATA CONTROL, I/O
C       CALLS:
C                   PAGEHD
C       CALLED BY:
C                   READ
C       PURPOSE:
C                   INITIALIZE TAPE FOR SUBROUTINE WTAPE
C
C***********************************************************
C
C     SUBROUTINE INVINP              DEBUG # 41
C                                             UTILITY, MATRIX OPERATION
C       CALLS:
C                   WRITES
C       CALLED BY:
C                   YDOT
C                   PRNTCD
C       PURPOSE:
```

```
C                    [illegible] MATRIX                                 013404
C                                                                       013405
C************************************************************           013406
C                                                                       013407
C     SUBROUTINE [illegible]            DEBUG # [illegible]             013408
C                                         USER DEFINED SUBROUTINE      013409
C        CALLS:                                                         013410
C                  [illegible]                                          013411
C        CALLED BY:                                                     013412
C                  [illegible]                                          013413
C        PURPOSE:                                                       013414
C                  USER DEFINED [illegible] FORCES AND                  013415
C                  [illegible] SPRINGS AND                              013416
C                  DAMPERS                                              013417
C                                                                       013418
C************************************************************           013419
C                                                                       013420
C     SUBROUTINE [illegible]            DEBUG # 40                      013421
C                                         [illegible] OPERATION         013422
C        CALLS:                                                         013423
C                  YDOT                                                 013424
C                  WRITES                                               013425
C        CALLED BY:                                                     013426
C                  SYNRLD                                               013427
C        PURPOSE:                                                       013428
C                  USED TO OBTAIN THE LINEARIZED EQUATIONS OF MOTION    013429
C                  USING THE THEORY DISCUSSED IN VOL I SECTION III      013430
C                  'SYNTHESIS AND ANALYSIS OF THE LINEARIZED SYSTEM'    013431
C                  THE PARTIAL DERIVATIVE MATRIX A IN                   013432
C                       YDOT(I) = A*Y                                   013433
C                  IS DERIVED AND WRITTEN ON NEAPLE                     013434
C                                                                       013435
C************************************************************           013436
C                                                                       013437
C     SUBROUTINE LPLTAP                 DEBUG # 100                     013438
C                                         UTILITY, DATA CONTROL, I/O    013439
C        CALLS:                                                         013440
C                  (NONE)                                               013441
C        CALLED BY:                                                     013442
C                  LTRESP                                               013443
C        PURPOSE:                                                       013444
C                  TO WRITE TAPE FOR PLOTTING LINEARIZED TIME           013445
C                  HISTORY RESPONSE DATA                                013446
C                                                                       013447
C************************************************************           013448
C                                                                       013449
C     SUBROUTINE LPRNT                  DEBUG # 101                     013450
C                                         UTILITY, DATA CONTROL, I/O    013451
C        CALLS:                                                         013452
C                  PAGEHD                                               013453
C                  WRITES                                               013454
C        CALLED BY:                                                     013455
C                  LTRESP                                               013456
C        PURPOSE:                                                       013457
C                  TO PRINT LINEARIZED TIME HISTORY RESPONSE            013458
C                                                                       013459
C************************************************************           013460
C                                                                       013461
C     SUBROUTINE LTAPE                  DEBUG # 14                      013462
C                                         UTILITY, DATA CONTROL, I/O    013463
```

```
C         CALLS:
C                        PAGEHD
C         CALLED BY:
C                        READ
C                        RTAPE
C         PURPOSE:
C                        LIST THE HEADINGS OF MATRICES WRITTEN ON
C                        TAPE NUMBER NTAPE
C
C***********************************************************
C
C      SUBROUTINE LTORQL             DEBUG # 103
C                                                USER DEFINED SUBROUTINE
C         CALLS:
C                        ZERO
C                        (OTHERS IF USER REQUESTED)
C         CALLED BY:
C                        LTRESP
C         PURPOSE:
C                        TO COMPUTE ANY EXTERNAL FORCES AND TORQUES ACTING
C                        ON THE SYSTEM FOR LINEARIZED TIME HISTORY ANALYSIS
C                        RESULTS TO BE PUT IN VTORQ BY USER. IN YOUTL DISCOS
C                        WILL THEN SET UP THE MATRIX EQUATIONS
C                              YDT = A*Y + VTORQ
C                        TO BE INTEGRATED IN LTRESP (NOTE UNITS OF VTORQ)
C
C***********************************************************
C
C      SUBROUTINE LTRESP             DEBUG # 99
C                                                MAIN LINE OPERATION
C         CALLS:
C                        ZERO       YOUTL      LTORQL
C                        MULTD      LPLTWR
C                        GAUSSI     LPRNT
C         CALLED BY:
C                        DYN540
C         PURPOSE:
C                        TO MAKE USE OF THE LINEARIZED EQUATIONS TO COMPUTE
C                        THE LINEARIZED TIME DOMAIN RESPONSE. SEE PAGES
C                        35-80 OF VOL I. OUTPUT DATA SENT TO BOTH
C                        PRINT (LPRNT) AND PLOT (LPLTWR) ROUTINES
C
C***********************************************************
C
C      MAIN  (DISCOS)                DEBUG # 1
C                                                MAIN LINE EXECUTIVE
C         CALLS:
C                        DYN510     START
C                        DYN520     COMENT
C                        DYN530
C                        DYN540
C         CALLED BY:
C                        NONE. THIS IS THE MAIN PROGRAM
C         PURPOSE:
C                        MAIN EXECUTIVE DRIVING PROGRAM. SIZES OF VARIOUS
C                        DIMENSIONED ARRAY DEFINED HERE
C                        THESE MAY BE CHANGED VIA THE REDIMENSION PROGRAM
C
C***********************************************************
C
```

```
C      SUBROUTINE MGEN                DEBUG # 49                                  013524
C                                              MAIN LINE OPERATION                013525
C         CALLS:                                                                  013526
C                       WRITES                                                    013527
C                       MULTS                                                     013528
C                       MULTAD                                                    013529
C         CALLED BY:                                                              013530
C                       YDOT                                                      013531
C         PURPOSE:                                                                013532
C                       TO COMPUTE THE DEFORMATION DEPENDENT MASS MATRIX          013533
C                                                                                 013534
C*****************************************************************               013535
C                                                                                 013536
C      SUBROUTINE MLTSR               DEBUG # 56                                  013537
C                                              UTILITY, MATRIX OPERATION          013538
C         CALLS:                                                                  013539
C                       (NONE)                                                    013540
C         CALLED BY:                                                              013541
C                       GETSMB                                                    013542
C         PURPOSE:                                                                013543
C                       USE TO PERFORM MATRIX OPERATION                           013544
C                             C = A*B                                             013545
C                       WHERE                                                     013546
C                          C(S,LM) = A(S,LE)*B(LE,LM)                             013547
C                       FOR SELECTED ROWS OF C   (NOTHING DESTROYED)              013548
C                                                                                 013549
C*****************************************************************               013550
C                                                                                 013551
C      SUBROUTINE RIGID               DEBUG # 28                                  013552
C                                              MAIN LINE OPERATION                013553
C         CALLS:                                                                  013554
C                       ZERO       WRITES     PAGEHD                              013555
C                       ROTTR      WRITIS                                         013556
C                       SKEWV3     READ                                           013557
C         CALLED BY:                                                              013558
C                       DYNSID                                                    013559
C         PURPOSE:                                                                013560
C                       READ AND PROCESS RIGID BODY DATA FOR ALL RIGID            013561
C                       BODIES, SEE FLOW CHART PAGE 24 VOL II, DATA INPUT         013562
C                       DEF3 AND DATA OUTPUT DEF4                                 013563
C                                                                                 013564
C*****************************************************************               013565
C                                                                                 013566
C      SUBROUTINE MSMODC              DEBUG # 15                                  013567
C                                              MAIN LINE OPERATION                013568
C         CALLS:                                                                  013569
C                       READ       SKEWV3     MULTA                               013570
C                       STASA      PRS        ROTTR                               013571
C                       UNITY      ZERO       READIM                              013572
C                       PAGSBT     PAGEHD     WRITIS                              013573
C                       REVISE     WRITES                                         013574
C         CALLED BY:                                                              013575
C                       DYNSID                                                    013576
C         PURPOSE:                                                                013577
C                       READ AND PROCESS FLEXIBLE BODY DATA FOR EACH              013578
C                       FLEXIBLE BODY DESCRIBED WITH CONSISTANT MASS DATA         013579
C                       SEE FLOW CHART PAGES 27-28 VOL II                         013580
C                        DATA INPUT SUB. DEF3, DATA OUTPUT SUB. DEF4,             013581
C                       THEORY APPENDIX A,VOL I                                   013582
C                                                                                 013583
```

```
C*************************************************************       013584
C                                                                    013585
C     SUBROUTINE MSMODL           DEBUG # 16                         013586
C                                        MAIN LINE OPERATION         013587
C        CALLS:                                                      013588
C                    CREE      CREA      FETCH                       013589
C                    READ      CRETS     CREC                        013590
C                    ROTTR     WRITE     ZERO                        013591
C                    CREADD    CREMOD    WRITES                      013592
C                    WRITIS    CREB                                  013593
C        CALLED BY:                                                  013594
C                    DYNSIO                                          013595
C        PURPOSE:                                                    013596
C                    READ AND PROCESS FLEXIBLE BODY DATA FOR EACH    013597
C                    FLEXIBLE BODY DESCRIBED WITH LUMPED MASS DATA   013598
C                    SEE FLOW CHART PAGES 25-26 VOL II, DATA INPUT   013599
C                    SUB. DEFS, DATA OUTPUT SUB. DEF., THEORY APPENDIX A  013600
C                    VOL I                                           013601
C                                                                    013602
C*************************************************************       013603
C                                                                    013604
C     SUBROUTINE MULTA            DEBUG # 34                         013605
C                                        UTILITY, MATRIX OPERATION   013606
C        CALLS:                                                      013607
C                    (NONE)                                          013608
C        CALLED BY:                                                  013609
C                    MSMODL                                          013610
C                    ASIMER                                          013611
C        PURPOSE:                                                    013612
C                    UTILITY ROUTINE TO MULTIPLY RECTANGULAR MATRICES  013613
C                              AZ * B = Z                            013614
C                    INPUT AZ AND B, OUTPUT Z IN DUMMY ARRAY AZ      013615
C                                                                    013616
C*************************************************************       013617
C                                                                    013618
C     SUBROUTINE MULTAD           DEBUG # 55                         013619
C                                        UTILITY, MATRIX OPERATION   013620
C        CALLS:                                                      013621
C                    (NONE)                                          013622
C        CALLED BY:                                                  013623
C                    AGEN      TORQUE                                013624
C                    GRVGRD    ENGMOM                                013625
C                    BODTGP                                          013626
C        PURPOSE:                                                    013627
C                    FORM MATRIX OPERATION                           013628
C                         ZZ = Z + A*B                               013629
C                    A,B,Z INPUT MATRICES                            013630
C                    ZZ OUTPUT IN Z ARRAY                            013631
C                    ORIGINAL Z DESTROYED A AND B ARE NOT            013632
C                                                                    013633
C*************************************************************       013634
C                                                                    013635
C     SUBROUTINE MULTB            DEBUG # 97                         013636
C                                        UTILITY, MATRIX OPERATION   013637
C        CALLS:                                                      013638
C                    (NONE)                                          013639
C        CALLED BY:                                                  013640
C                    ASIMER                                          013641
C                    ETRGSP                                          013642
C        PURPOSE:                                                    013643
```

```
C                 UTILITY ROUTINE TO MULTIPLY RECTANGULAR MATRICES        013644
C                         A * B2 = Z                                      013645
C                 INPUT A AND B2, OUTPUT Z IN THE DUMMY ARRAY B2          013646
C                                                                         013647
C**************************************************************           013648
C                                                                         013649
C     SUBROUTINE [illegible]        [illegible] # 54                      013650
C                                         UTILITY, MATRIX OPERATION       013651
C         CALLS:                                                          013652
C                 [illegible]                                             013653
C         CALLED BY:                                                      013654
C                 [illegible]     [illegible]     [illegible]             013655
C                 [illegible]     [illegible]     [illegible]             013656
C                 [illegible]     [illegible]     [illegible]             013657
C                 [illegible]     [illegible]                             013658
C         PURPOSE:                                                        013659
C                 FORM GENERAL MATRIX PRODUCT OF RECTANGULAR MATRICES     013660
C                 [illegible]                                             013661
C                 INPUT A AND B2, OUTPUT Z [illegible] STORED             013662
C                                                                         013663
C**************************************************************           013664
C                                                                         013665
C     SUBROUTINE NICPLT             [illegible] # 93                      013666
C                                         UTILITY, DATA PROCESSING        013667
C                                         UTILITY, DATA CONTROL, I/O      013668
C         CALLS:                                                          013669
C                 (SC4020 PLOT ROUTINES)                                  013670
C         CALLED BY:                                                      013671
C                 DYN340                                                  013672
C         PURPOSE:                                                        013673
C                 GENERATE NICHOLS PLOTS AMPLITUDE VS PHASE ON            013674
C                 RECTANGULAR GRID, DATA IN [illegible]                   013675
C                                                                         013676
C**************************************************************           013677
C                                                                         013678
C     SUBROUTINE [illegible]        [illegible] # 13                      013679
C                                         MA[illegible] OPERATION         013680
C         CALLS:                                                          013681
C                 [illegible]     [illegible]                             013682
C                 [illegible]                                             013683
C                 SIFT                                                    013684
C         CALLED BY:                                                      013685
C                 DYN340                                                  013686
C         PURPOSE:                                                        013687
C                 MAKE USE OF THE REDUCED CHARACTERISTIC MATRIX AND       013688
C                 CORRESPONDING INPUT COEFFICIENTS TO SET UP THE          013689
C                 NUMERATOR FOR THE PARTICULAR TRANSFER FUNCTION          013690
C                 REQUESTED AND TO CALL THE ROUTINES TO GET ITS ROOTS     013691
C                                                                         013692
C**************************************************************           013693
C                                                                         013694
C     SUBROUTINE NYPLT              [illegible] # 91                      013695
C                                         UTILITY, DATA PROCESSING        013696
C                                         UTILITY, DATA CONTROL, I/O      013697
C         CALLS:                                                          013698
C                 PLOTSS                                                  013699
C                 (SC4020 PLOT ROUTINES)                                  013700
C         CALLED BY:                                                      013701
C                 DYN340                                                  013702
C         PURPOSE:                                                        013703
```

```
C                 GENERATE NYQUIST PLOTS GAIN VS PHASE ON                 013704
C                 POLAR COORDINATE GRID, DATA IN /PSTUFF/                 013705
C                                                                         013706
C************************************************************            013707
C                                                                         013708
C      SUBROUTINE PAGEHD            DEBUG # 8                             013709
C                                          UTILITY, DATA CONTROL, I/O     013710
C         CALLS:                                                          013711
C                 TIME                                                    013712
C                 SECOND                                                  013713
C         CALLED BY:                                                      013714
C                 DYNS10    READ      PRNTOU    SFREQ2                    013715
C                 READIM    WRITE     DYNS30    REDCOS                    013716
C                 WRITIM    LTAPE     DYNS40    RWRITE                    013717
C                 COMENT    ASMODC    FINDT     LPRNT                     013718
C                 INTAPE    WRIGID    TFPLY                               013719
C         PURPOSE:                                                        013720
C                 UTILITY ROUTINE, BRINGS UP NEW PAGE ON LINE-PRINTER     013721
C                 AND PUTS HEADING AT TOP OF IT                           013722
C                                                                         013723
C************************************************************            013724
C                                                                         013725
C      SUBROUTINE PLOTSS            DEBUG # 94                            013726
C                                          UTILITY, DATA PROCESSING       013727
C         CALLS:                                                          013728
C                 (NONE)                                                  013729
C         CALLED BY:                                                      013730
C                 REPLOT                                                  013731
C                 NYPLOT                                                  013732
C         PURPOSE:                                                        013733
C                 SELECTS PLOT UPPER AND LOWER LIMITS FOR A 10 SQUARE     013734
C                 LINEAR PLOT GRID                                        013735
C                                                                         013736
C************************************************************            013737
C                                                                         013738
C     SUBROUTINE PLOTWR             DEBUG # 61                            013739
C                                          UTILITY, DATA CONTROL, I/O     013740
C         CALLS:                                                          013741
C                 (NONE)                                                  013742
C         CALLED BY:                                                      013743
C                 DYNS20                                                  013744
C         PURPOSE:                                                        013745
C                 PUT ALL TIME HISTORY DATA ON NTAPE3, PLOTS MAY BE       013746
C                 MADE OF ANY OF THE VARIABLES WRITTEN ON NTAPE3          013747
C                                                                         013748
C************************************************************            013749
C                                                                         013750
C      SUBROUTINE PLTCAR            DEBUG # 63                            013751
C                                          UTILITY, DATA CONTROL, I/O     013752
C         CALLS:                                                          013753
C                 RSETMG    SETSMG    SETUPG                              013754
C                 TITLEG    GRIDG     LINESG                              013755
C                 PAGEG     TEXTG     SUBJEG                              013756
C                 LABELG    LEGNDG                                        013757
C                 NUMBRG    OBJCTG                                        013758
C         CALLED BY:                                                      013759
C                 DYNS50                                                  013760
C         PURPOSE:                                                        013761
C                 MAKES USE OF SC4020 SUBROUTINES WHICH ARE RESIDENT      013762
C                 IN THE GSFC IBM-360 SYSTEM TO GENERATE PLOTS FOR        013763
```

```
C                   OUTPUT DATA GENERATED VIA NON-LINEAR TIME RESPONSE     013764
C                   ANALYSIS. THE DATA IS TAKEN OFF NTAPES IN DYNS30       013765
C                   PUT IN DUMMY ARRAY AND PASSED TO PLTCAR FOR PLOT       013766
C                   GENERATION                                             013767
C                                                                          013768
C*****************************************************************         013769
C                                                                          013770
C     SUBROUTINE PRNTDD             DEBUG # 60                             013771
C                                           UTILITY, DATA CONTROL, I/O     013772
C        CALLS:                                                            013773
C                   PAGEHD     INVINP                                      013774
C                   WRITES                                                 013775
C                   SECOND                                                 013776
C        CALLED BY:                                                        013777
C                   DYNS30                                                 013778
C        PURPOSE:                                                          013779
C                   GENERAL PRINT ROUTINE FOR NON-LINEAR TIME HISTORY      013780
C                   RESPONSE                                               013781
C                                                                          013782
C*****************************************************************         013783
C                                                                          013784
C     SUBROUTINE PR3                DEBUG # 36                             013785
C                                           UTILITY, MATRIX OPERATION      013786
C        CALLS:                                                            013787
C                   (NONE)                                                 013788
C        CALLED BY:                                                        013789
C                   ASMODC                                                 013790
C        PURPOSE:                                                          013791
C                   TO FORM THE MATRIX PRODUCT                             013792
C                     ZZ = Z + S*C**T*A*B                                  013793
C                   WHERE                                                  013794
C                   Z IS NCC,NCB  INPUT DESTROYED, OUTPUT RETURNED HERE    013795
C                   C IS NRA,NCC  NOT DESTROYED                            013796
C                   A IS NRA,NCA  NOT DESTROYED                            013797
C                   B IS NCA,NCB  NOT DESTROYED                            013798
C                   S IS SCALAR   NOT DESTROYED                            013799
C                                                                          013800
C*****************************************************************         013801
C                                                                          013802
C     SUBROUTINE QRCON              DEBUG # 85                             013803
C                                           UTILITY, MATRIX OPERATION      013804
C        CALLS:                                                            013805
C                   QR2                                                    013806
C        CALLED BY:                                                        013807
C                   QRDRVR                                                 013808
C        PURPOSE:                                                          013809
C                   TO CALL FOR THE QR TRANSFORMATION AND TO MAKE USE      013810
C                   OF THE DATA CORRECTED THEREIN TO DEFINE THE ROOTS OF   013811
C                   THE MATRIX (REAL ROOTS OR/REAL,COMPLEX PAIRS)          013812
C                                                                          013813
C*****************************************************************         013814
C                                                                          013815
C     SUBROUTINE QRDRVR             DEBUG # 34                             013816
C                                           UTILITY, MATRIX OPERATION      013817
C        CALLS:                                                            013818
C                   QRCON                                                  013819
C                   SORTEA                                                 013820
C                   WRITES                                                 013821
C        CALLED BY:                                                        013822
C                   DYNS30                                                 013823
```

```
C                   NUMS                                                013824
C       PURPOSE:                                                        013825
C                   TO DO A MINIMAL AMOUNT OF DATA ORGANIZATION AND     013826
C                   TO CALL THE ROUTINES THAT WILL COMPUTED THE COMPLEX 013827
C                   EIGENVALUES OF THE N,N MATRIX A. THIS ROUTINE IS    013828
C                   AVAILABLE IN EXTENDED REAL*16 PRECISION HOWEVER     013829
C                   RUN TIME ON IBM-360 IS ORDERS OF MAGNITUDE SLOWER   013830
C                   THAN REAL*8 PRECISION                               013831
C                                                                       013832
C***********************************************************            013833
C                                                                       013834
C     SUBROUTINE QR2                DEBUG # 87                          013835
C                                          UTILITY, MATRIX OPERATION    013836
C       CALLS:                                                          013837
C                   (NONE)                                              013838
C       CALLED BY:                                                      013839
C                   QRCON                                               013840
C       PURPOSE:                                                        013841
C                   TO GO THROUGH THE QR-TRANSFORMATION AND RETURN TO   013842
C                   QRCON THE DATA NECESSARY FOR SPECIFICATION OF THE   013843
C                   COMPLEX ROOTS                                       013844
C                                                                       013845
C***********************************************************            013846
C                                                                       013847
C     SUBROUTINE READ               DEBUG # 10                          013848
C                                          UTILITY, DATA CONTROL, I/O   013849
C       CALLS:                                                          013850
C                   INTAPE    WRITE                                     013851
C                   KTAPE     LTAPE                                     013852
C                   PAGEHD    WTAPE                                     013853
C       CALLED BY:                                                      013854
C                   DYNS10    NSMODE    DYNS40                          013855
C                   MSMODE    MRIGID    CONTRL                          013856
C       PURPOSE:                                                        013857
C                   READ MATRIX OF REAL NUMBERS FROM CARDS OR TAPE,     013858
C                   AND PRINT IT, SEE COMMENT CARDS IN SUBROUTINE       013859
C                                                                       013860
C***********************************************************            013861
C                                                                       013862
C     SUBROUTINE READER             DEBUG # 3                           013863
C                                          UTILITY, DATA CONTROL, I/O   013864
C       CALLS:                                                          013865
C                   PAGEHD                                              013866
C       CALLED BY:                                                      013867
C                   DYNS10    DYNS40                                    013868
C                   MSMODE                                              013869
C       PURPOSE:                                                        013870
C                   READ INTEGER MATRIX IN FORMA FORMAT FROM CARDS      013871
C                   PRINT ECHO OF INPUT IF * IN COL 80 OF HEADER CARD   013872
C                                                                       013873
C***********************************************************            013874
C                                                                       013875
C     SUBROUTINE REVADD             DEBUG # 98                          013876
C                                          UTILITY, MATRIX OPERATION    013877
C       CALLS:                                                          013878
C                   (NONE)                                              013879
C       CALLED BY:                                                      013880
C                   FINDT                                               013881
C                   ASTMER                                              013882
C       PURPOSE:                                                        013883
```

```
C                  [illegible]                                  013884
C                         [illegible]                           013885
C                                                               013886
C**************************************************************  013887
C                                                               013888
C        [illegible]                          [illegible]       013889
C                                   UTILITY, MATRIX OPERATION   013890
C        [illegible]:                                           013891
C                 [illegible]                                   013892
C        [illegible]:                                           013893
C                 [illegible]                                   013894
C        [illegible]:                                           013895
C                 [illegible]                                   013896
C                 [illegible]                                   013897
C                                                               013898
C**************************************************************  013899
                                                                013900
C        [illegible]                          [illegible]       013901
C                                   UTILITY, NON-MATRIX OPER.   013902
C        [illegible]:                                           013903
C                 [illegible]                                   013904
C                 [illegible]                                   013905
C                 [illegible]                                   013906
C        CALLED BY:                                             013907
C                 [illegible]                                   013908
C        PURPOSE:                                               013909
C                 [illegible] TRANSFER                          013910
C                 [illegible] AND SAVER                         013911
C                 [illegible]                                   013912
C                                                               013913
C**************************************************************  013914
C                                                               013915
C        SUBROUTINE RKADAM            DEPGS # 59                013916
C                                   UTILITY, NON-MATRIX OPER.   013917
C        CALLS:                                                 013918
C                 ROOT                                          013919
C        CALLED BY:                                             013920
C                 [illegible]                                   013921
C        PURPOSE:                                               013922
C                 BASIC INTEGRATION PACKAGE. EITHER FOURTH ORDER RUNGE-  013923
C                 KUTTA OR ADAMS PREDICTOR CORRECTOR METHODS AVAILABLE   013924
C                                                               013925
C**************************************************************  013926
C                                                               013927
C        SUBROUTINE [illegible]       DEPGS # [illegible]       013928
C                                   UTILITY, DATA PROCESSING    013929
C                                   UTILITY, DATA CONTROL, I/O  013930
C        CALLS:                                                 013931
C                 PLOTS                                         013932
C                 (SC4020 PLOT ROUTINES)                        013933
C        CALLED BY:                                             013934
C                 [illegible]                                   013935
C        PURPOSE:                                               013936
C                 GENERATE ROOT LOCI PLOT FOR A SINGLE ROOT     013937
C                 DATA IN /RSTUFF/                              013938
C                                                               013939
C**************************************************************  013940
C                                                               013941
C     SUBROUTINE [illegible]          DEPGS # 71                013942
C                                   UTILITY, NON-MATRIX OPER.   013943
```

```
C         CALLS:                                                        013944
C                      (NONE)                                           013945
C         CALLED BY:                                                    013946
C                      DYNS4D                                           013947
C        PURPOSE:                                                       013948
C                      REMOVE REAL ZEROS FROM REAL YDOT ARRAY PRIOR     013949
C                      TO CALL TO TIFF                                  013950
C                                                                       013951
C**********************************************************             013952
C                                                                       013953
C      SUBROUTINE ROTOM           DEBUG # 44                            013954
C                                          MAIN LINE OPERATION          013955
C         CALLS:                                                        013956
C                      MULTS                                            013957
C                      ROTTR                                            013958
C                      WRITES                                           013959
C         CALLED BY:                                                    013960
C                      DYNS2D                                           013961
C                      YDOT                                             013962
C         PURPOSE:                                                      013963
C                      TO COMPUTE HINGE FRAME TRANSFORMATIONS AND VELOCITY  013964
C                      TRANSFORMATION MATRICES. THE ELEMENTS OF THE ARRAYS  013965
C                      OH,RH,PIN,RGL AND DGL ARE COMPUTED HERE          013966
C                                                                       013967
C**********************************************************             013968
C                                                                       013969
C      SUBROUTINE ROTOS           DEBUG # 46                            013970
C                                          MAIN LINE OPERATION          013971
C         CALLS:                                                        013972
C                      MULTS                                            013973
C                      ROTTR                                            013974
C         CALLED BY:                                                    013975
C                      YDOT                                             013976
C         PURPOSE:                                                      013977
C                      TO COMPUTE SENSOR POINT FRAME TRANSFORMATION MATRIX  013978
C                      THE ELEMENTS OF THE ARRAYS RS AND QS ARE COMPUTED  013979
C                      HERE                                             013980
C                                                                       013981
C**********************************************************             013982
C                                                                       013983
C      SUBROUTINE ROTTR           DEBUG # 32                            013984
C                                          MAIN LINE OPERATION          013985
C         CALLS:                                                        013986
C                      MULTS                                            013987
C         CALLED BY:                                                    013988
C                      MSMODE     ROTOM                                 013989
C                      MSMODE     ROTOS                                 013990
C                      MRESID                                           013991
C         PURPOSE:                                                      013992
C                      TO COMPUTE ANY ONE OF THE TWELVE TYPES OF EULER  013993
C                      ANGLE TRANSFORMATION MATRICES AND/OR THE PI INVERSE  013994
C                      VELOCITY TRANSFORMATIONS. FOR THEORY SEE APPENDIX B,  013995
C                      VOL I. ALSO SEE COMMENT CARDS ROTTR              013996
C                                                                       013997
C**********************************************************             013998
C                                                                       013999
C      SUBROUTINE RTAPE           DEBUG # 11                            014000
C                                          UTILITY, DATA CONTROL, I/O   014001
C         CALLS:                                                        014002
C                      LTAPE                                            014003
```

```
C        CALLED BY:                                                      014004
C                  READ                                                  014005
C        PURPOSE:                                                        014006
C                  READ MATRIX A, SIZE(NRA,NCA) FROM TAPE ALONG WITH     014007
C                  ROW AND COLUMN LENGTHS. SEE COMMENT CARDS IN RTAPE    014008
C                                                                        014009
C*****************************************************************        014010
C                                                                        014011
C      SUBROUTINE RTOP              DEBUG # 81                           014012
C                                              UTILITY, NON-MATRIX OPER. 014013
C        CALLS:                                                          014014
C                  (NONE)                                                014015
C        CALLED BY:                                                      014016
C                  DYN340                                                014017
C        PURPOSE:                                                        014018
C                  TO TAKE THE NUMERATOR AND DENOMINATOR ROOTS STORED    014019
C                  IN THE ROOTS ARRAY R AND GENERATE THE CORRESPONDING   014020
C                  POLYNOMIAL COEFFICIENTS. THE COEFFICIENTS ARE         014021
C                  STORED IN ARRAY RX OF /ZEROS/                         014022
C                                                                        014023
C*****************************************************************        014024
C                                                                        014025
C      SUBROUTINE RWRITE            DEBUG # 88                           014026
C                                              UTILITY, DATA CONTROL, I/O 014027
C        CALLS:                                                          014028
C                  PAGEHD                                                014029
C        CALLED BY:                                                      014030
C                  DYN340                                                014031
C        PURPOSE:                                                        014032
C                  PRINT OUT REAL AND IMAGINARY PARTS OF THE ROOTS       014033
C                  OBTAINED AT DIFFERENT STEPS THROUGH THE STABILITY     014034
C                  ANALYSIS                                              014035
C                                                                        014036
C*****************************************************************        014037
C                                                                        014038
C      SUBROUTINE SATB              DEBUG # 26                           014039
C                                              UTILITY, MATRIX OPERATION 014040
C        CALLS:                                                          014041
C                  (NONE)                                                014042
C        CALLED BY:                                                      014043
C                  [illegible]    [illegible]                            014044
C                  [illegible]    [illegible]                            014045
C                  [illegible]    [illegible]                            014046
C        PURPOSE:                                                        014047
C                  PERFORM THE MATRIX OPERATION                          014048
C                    ZZ = Z + S*AT*B                                     014049
C                  Z,A,B   INPUT MATRICES                                014050
C                  S       INPUT SCALAR                                  014051
C                  ZZ      OUTPUT MATRIX RETURNED IN Z ARRAY             014052
C                  Z INPUT DESTROYED                                     014053
C                  A,B NOT DESTROYED                                     014054
C                                                                        014055
C*****************************************************************        014056
C                                                                        014057
C      SUBROUTINE SCALE             DEBUG # 33                           014058
C                                              UTILITY, DATA PROCESSING  014059
C        CALLS:                                                          014060
C                  (NONE)                                                014061
C        CALLED BY:                                                      014062
C                  [illegible]                                           014063
```

```
C         PURPOSE:                                                      014064
C                   COMPUTE SCALE FACTOR AND SCALE LIMITS FOR PLOTTING  014065
C                                                                       014066
C*****************************************************************      014067
C                                                                       014068
C      SUBROUTINE SFEQZ            DEBUG # 73                           014069
C                                          UTILITY, NON-MATRIX OPER.    014070
C         CALLS:                                                        014071
C                   MAGEND    DREAL                                     014072
C                   DIMAG                                               014073
C         CALLED BY:                                                    014074
C                   DYNSYD                                              014075
C         PURPOSE:                                                      014076
C                   TO DETERMINE S-PLANE FREQUENCY RESPONSE. ALL DATA   014077
C                   COMPUTED STORED IN /PSTUFF/ FOR PLOTTING            014078
C                   MAKES USE OF TIME CONSTANTS, DAMPING ZETAS AND      014079
C                   FREQUENCIES STORED IN ARRAYS FDRN,FDNC,FER=FBRD     014080
C                   AND FBDE ALONG WITH BODE GAIN=GAIN                  014081
C                                                                       014082
C*****************************************************************      014083
C                                                                       014084
C      SUBROUTINE SHAFT            DEBUG # 109                          014085
C                                          USER DEFINED SUBROUTINE      014086
C         CALLS:                                                        014087
C                   (NONE UNLESS USER REQUESTED)                        014088
C         CALLED BY:                                                    014089
C                   TORQUE                                              014090
C         PURPOSE:                                                      014091
C                   TO DEFINE ALL TORQUES ACTING ON EMBEDDED            014092
C                   SYMMETRIC WHEELS (USER DEFINED)                     014093
C                                                                       014094
C*****************************************************************      014095
C                                                                       014096
C      SUBROUTINE SIFT             DEBUG # 72                           014097
C                                          UTILITY, NON-MATRIX OPER.    014098
C         CALLS:                                                        014099
C                   (NONE)                                              014100
C         CALLED BY:                                                    014101
C                   DYNSYD                                              014102
C                   NOTS                                                014103
C         PURPOSE:                                                      014104
C                   SEARCH INPUT ARRAY A FOR NUMBERS LESS THAN TOLERANCE 014105
C                   VALUE TOL, SET THESE EQUAL TO 0.0                   014106
C                                                                       014107
C*****************************************************************      014108
C                                                                       014109
C      SUBROUTINE SKEWV            DEBUG # 55                           014110
C                                          UTILITY, MATRIX OPERATION    014111
C         CALLS:                                                        014112
C                   (NONE)                                              014113
C         CALLED BY:                                                    014114
C                   ASMODE    [illegible]                               014115
C                   ARESID    [illegible]                               014116
C                   DMODEN    TORQUE                                    014117
C         PURPOSE:                                                      014118
C                   UTILITY ROUTINE SKEW MATRIX                         014119
C                                 0    V3  -V2                          014120
C                         SKV = -V3    0    V1                          014121
C                                V2  -V1    0                           014122
C                   INPUT (V1,V2,V3)   OUTPUT SKV ARRAY                 014123
```

```
C                                                                       014124
C*************************************************************          014125
C                                                                       014126
C     SUBROUTINE [illegible]                 DEBUG # [illegible]        014127
C                                          UTILITY, DATA PROCESSING     014128
C                                          UTILITY, DATA CONTROL, I/O   014129
C        CALLS:                                                         014130
C                    [illegible]                                        014131
C                 [illegible]                                           014132
C        CALLED BY:                                                     014133
C                    [illegible]                                        014134
C        PURPOSE:                                                       014135
C                 [illegible] EFFICIENCY                                014136
C                 [illegible]                                           014137
C                                                                       014138
C*************************************************************          014139
C                                                                       014140
C     SUBROUTINE [illegible]                 DEBUG # [illegible]        014141
C                                          UTILITY, DATA CONTROL, I/O   014142
C        CALLS:                                                         014143
C                 [illegible]                                           014144
C                 [illegible]                                           014145
C                 [illegible]                                           014146
C        CALLED BY:                                                     014147
C                 MAIN                                                  014148
C        PURPOSE:                                                       014149
C                 INITIALIZE [illegible] CAPABILITY, [illegible] DATA TO BE PUT  014150
C                 [illegible]                                           014151
C                 [illegible] SUBROUTINES                               014152
C                 [illegible]                                           014153
C                 [illegible]                                           014154
C                                                                       014155
C*************************************************************          014156
C                                                                       014157
C     SUBROUTINE [illegible]                 DEBUG # [illegible]        014158
C                                          UTILITY, MATRIX OPERATION    014159
C                                          UTILITY, DATA CONTROL, I/O   014160
C        CALLS:                                                         014161
C                 (NONE)                                                014162
C        CALLED BY:                                                     014163
C                 [illegible]                                           014164
C        PURPOSE:                                                       014165
C                 FORM MATRIX PRODUCT                                   014166
C                       C = A*B                                         014167
C                 WHERE                                                 014168
C                    A IS (NA,NA) DIAGONAL STORED AS A VECTOR           014169
C                    B IS (NA,NCB) GENERAL                              014170
C                 WRITE C IN BINARY ON TAPE NTAPE                       014171
C                                                                       014172
C*************************************************************          014173
C                                                                       014174
C     SUBROUTINE [illegible]DIA              DEBUG # 06                 014175
C                                          UTILITY, MATRIX OPERATION    014176
C        CALLS:                                                         014177
C                 (NONE)                                                014178
C        CALLED BY:                                                     014179
C                 [illegible]                                           014180
C        PURPOSE:                                                       014181
C                 TO PUT THE MATRIX A IN UPPER HESSENBERG FORM,         014182
C                 PRELIMINARY STEP IN OBTAINING COMPLEX ROOTS           014183
```

```
C                                                                       014184
C***********************************************************            014185
C                                                                       014186
C     SUBROUTINE TFPOLY           DEBUG # 69                            014187
C                                            MAIN LINE OPERATION        014188
C        CALLS:                                                         014189
C                   MAGIND                                              014190
C        CALLED BY:                                                     014191
C                   CONTRL                                              014192
C        PURPOSE:                                                       014193
C                   TO CONVERT POLYNOMIAL TRANSFER FUNCTION EXPRESSIONS 014194
C                   OF CONTROL LOOPS TO FIRST ORDER DIFFERENTIAL        014195
C                   EQUATIONS. SEE VOL I PAGE 73 FOR THEORY.            014196
C                   OUTPUT OF TFPOLY LOADED DIRECTLY INTO STATE VECTOR  014197
C                   DERIVATIVE ARRAY YDT                                014198
C                                                                       014199
C***********************************************************            014200
C                                                                       014201
C     SUBROUTINE TFTYPE           DEBUG # 67                            014202
C                                            MAIN LINE OPERATION        014203
C        CALLS:                                                         014204
C                   ZERO                                                014205
C        CALLED BY:                                                     014206
C                   DYNSAD                                              014207
C        PURPOSE:                                                       014208
C                   MAKES USE OF THE INPUTTED INTEGER CODES AND THE     014209
C                   LINEARIZED SYSTEM EQUATIONS OF MOTION TO SET UP     014210
C                   THE REDUCED CHARACTERISTIC MATRIX AND THE           014211
C                   CORRESPONDING INPUT COEFFICIENTS FOR THE PARTICULAR 014212
C                   TRANSFER FUNCTION REQUESTED. FOR THEORY SEE VOL 1.  014213
C                   PAGES 73-76                                         014214
C                                                                       014215
C***********************************************************            014216
C                                                                       014217
C     SUBROUTINE TORQUE           DEBUG # 57                            014218
C                                            MAIN LINE OPERATION        014219
C        CALLS:                                                         014220
C                   MULTS    CONTRL   WRITES     SNEWV3                 014221
C                   MULTAD   EXTOR    SMISC                             014222
C                   EQADD    SHAFFT   RHINGE                            014223
C        CALLED BY:                                                     014224
C                   YDOT                                                014225
C        PURPOSE:                                                       014226
C                   COMPUTE ALL GYROSCOPIC FORCES AND TORQUES ACTING    014227
C                   ON THE SYSTEM, CALL USER ROUTINES WHICH DEFINE      014228
C                   NON-GYROSCOPIC EFFECTS (SPRINGS, DASHPOTS, MOTORS)  014229
C                   WHEEL CONTROL TORQUES, GRAVITY GRADIENT, ALSO       014230
C                   CALL TO EQADD DEFINES THE SENSOR AND TORQUE SIGNAL  014231
C                   EQUATIONS FOR STABILITY ANALYSIS                    014232
C                                                                       014233
C                                                                       014234
C***********************************************************            014235
C                                                                       014236
C     SUBROUTINE TREB             DEBUG # 82                            014237
C        CALLS:                                                         014238
C                   (NONE)                                              014239
C        CALLED BY:                                                     014240
C                   (NONE)                                              014241
C        PURPOSE :                                                      014242
C                                                                       014243
```

```
C*****************************************************               014244
C                                                                    014245
C      SUBROUTINE TIFF              DEBUG # 33                       014246
C                                        UTILITY, NON-MATRIX OPER.   014247
C         CALLS:                                                     014248
C                   (NONE)                                           014249
C         CALLED BY:                                                 014250
C                   DYNS4D                                           014251
C         PURPOSE:                                                   014252
C                   TAKES ALL REAL,ZERO AND COMPLEX ROOTS OF NUMERATOR   014253
C                   AND DENOMINATOR ALONG WITH BODE GAIN FOR PARTICULAR  014254
C                   TRANSFER FUNCTION AND PLACES THEM IN THE ROOTS       014255
C                   ARRAY R                                          014256
C                                                                    014257
C*****************************************************               014258
C                                                                    014259
C      SUBROUTINE UNITY             DEBUG # 33                       014260
C                                        UTILITY, MATRIX OPERATION   014261
C         CALLS:                                                     014262
C                   (NONE)                                           014263
C         CALLED BY:                                                 014264
C                   RSMODE                                           014265
C         PURPOSE:                                                   014266
C                   CREATE UNIT DIAGONAL MATRIX                      014267
C                                                                    014268
C*****************************************************               014269
C                                                                    014270
C      SUBROUTINE [illegible]       DEBUG # 10                       014271
C                                        UTILITY, DATA CONTROL, I/O  014272
C         CALLS:                                                     014273
C                   PAGEHD                                           014274
C         CALLED BY:                                                 014275
C                   [illegible]                                      014276
C                   [illegible]                                      014277
C                   [illegible]                                      014278
C                   [illegible]                                      014279
C         PURPOSE:                                                   014280
C                   UTILITY ROUTINE USED TO PRINT TWO DIMENSIONAL MATRIX 014281
C                   OF REAL NUMBERS IN [illegible] FORMAT WITH CAPTION HEADING 014282
C                   AT TOP OF NEW PAGE                               014283
C                                                                    014284
C*****************************************************               014285
C                                                                    014286
C      SUBROUTINE [illegible]       DEBUG # 43                       014287
C                                        UTILITY, DATA CONTROL, I/O  014288
C         CALLS:                                                     014289
C                   (NONE)                                           014290
C         CALLED BY:                                                 014291
C                   [illegible]                                      014292
C                   [illegible]                                      014293
C                   [illegible]                                      014294
C                   [illegible]                                      014295
C                   [illegible]                                      014296
C                   [illegible]                                      014297
C         PURPOSE:                                                   014298
C                   UTILITY [illegible] TO PRINT TWO DIMENSIONAL ARRAY OF 014299
C                   [illegible] FORMAT                               014300
C                                                                    014301
C*****************************************************               014302
                                                                     014303
```

```
C      SUBROUTINE WRITIM            DEBUG # 4                                 014304
C                                            UTILITY, DATA CONTROL, I/O  014305
C         CALLS:                                                          014306
C                   PAGEHD                                                014307
C         CALLED BY:                                                      014308
C                   (NONE)                                                014309
C         PURPOSE:                                                        014310
C                   PULL UP NEW PAGE PRINT INTEGER ARRAY IN FORMA         014311
C                   FORMAT WITH LABEL                                     014312
C                                                                         014313
C*****************************************************************        014314
C                                                                         014315
C      SUBROUTINE WRITIS            DEBUG # 5                             014316
C                                            UTILITY, DATA CONTROL, I/O  014317
C         CALLS:                                                          014318
C                   (NONE)                                                014319
C         CALLED BY:                                                      014320
C                   DYNS1D    DYNS2D    ASIMER                            014321
C                   MSMODC    FINDD                                       014322
C                   MSMODE    ROTDM                                       014323
C                   MRIGID    FINDT                                       014324
C         PURPOSE:                                                        014325
C                   PRINT TWO DIMENSIONAL INTEGER ARRAYS IN FORMA FORMAT  014326
C                                                                         014327
C*****************************************************************        014328
C                                                                         014329
C      SUBROUTINE WTAPE             DEBUG # 13                            014330
C                                            UTILITY, DATA CONTROL, I/O  014331
C         CALLS:                                                          014332
C                   (NONE)                                                014333
C         CALLED BY:                                                      014334
C                   READ                                                  014335
C         PURPOSE:                                                        014336
C                   WRITE MATRIX A IN BINARY ON TAPE, SEE COMMENT         014337
C                   CARDS IN WTAPE                                        014338
C                                                                         014339
C*****************************************************************        014340
C                                                                         014341
C      SUBROUTINE YDOT              DEBUG # 39                            014342
C                                            MAIN LINE OPERATION          014343
C         CALLS:                                                          014344
C                   ADJT      DDCTGP    HFGENR    DAKSEV                  014345
C                   DECMP     GETGAG    GRVGRD    DSGENR                  014346
C                   FINDD     WRITIS    MGEN      INVIMP                  014347
C                   ROTDS     MULT3     ROTDM     TARGSE                  014348
C         CALLED BY:                                                      014349
C                   DYNS2D                                                014350
C                   KRADAM                                                014351
C                   LINEAR                                                014352
C         PURPOSE:                                                        014353
C                   TO COMPUTE THE FIRST TIME DERIVATIVE OF THE           014354
C                   STATE VECTOR Y, SEE COMMENT CARDS IN YDOT             014355
C                                                                         014356
C*****************************************************************        014357
C                                                                         014358
C      SUBROUTINE YDOTE             DEBUG # 102                           014359
C                                            MAIN LINE OPERATION          014360
C         CALLS:                                                          014361
C                   (NONE)                                                014362
C         CALLED BY:                                                      014363
```

```
C              [illegible]                                              014364
C     PURPOSE:                                                          C14365
C              [illegible] GENERATE [illegible] COMBINATIONS OF         C14366
C              [illegible] OF THE FORM                                  014367
C                    [illegible]                                        C14368
C              [illegible] STORED IN [illegible]                        014369
C                                                                       C14370
C*********************************************************************  C14371
C                                                                       014372
C     ROUTINE [illegible]                 DEBUG # [illegible]           C14373
C                                 UTILITY, MATRIX OPERATION             C14374
C     CALLS:                                                            014375
C              (NONE)                                                   C14376
C     CALLED BY:                                                        014377
C              [illegible]  [illegible]    FINDT      [illegible]       C14378
C              [illegible]  [illegible]    TETYPE                       014379
C              [illegible]  [illegible]    [illegible]                  C14380
C              [illegible]  [illegible]    [illegible]                  C14381
C     PURPOSE:                                                          C14382
C              UTILITY ROUTINE TO ENTER 0.0 IN THE FIRST NR ROWS        C14383
C              AND [illegible] OF THE DUMMY ARRAY [illegible]           014384
C                                                                       C14385
C*********************************************************************  C14386
C                                                                       C14387
C     SUBROUTINES [illegible] TIME, AND SECOND                          C14388
C       PURPOSE:                                                        014389
C              SYSTEM ROUTINES FOR DEFINING [illegible] TIME            C14390
C              DATE AND TIME USED TO FIND ACTUAL TIME AND DATE          C14391
C              OF RUN.                                                  C14392
C              SECOND USED TO FIND ELAPSED CPU TIME AT VARIOUS          014393
C                [illegible] IN PROGRAM                                 014394
C                                                                       014395
C*********************************************************************  014396
C                                                                       014397
C     SECTION [illegible]                                               C14398
C       PURPOSE:                                                        014399
C              [illegible] SYSTEM PLOTTING ROUTINES. THESE ARE RESIDENT 014400
C              IN THE [illegible] SYSTEM                                C14401
C                                                                       C14402
C*********************************************************************  C14403
C                                                                       014404
C                                                                       C14405
C                                                                       C14406
C                                                                       C14407
C                                                                       C14408
C                                                                       014409
C                                                                       014410
      RETURN                                                            C14411
      END                                                               C14412





      SUBROUTINE [illegible]                                            C14413
C                                                       DEBUG # 114     014414
C                                                                       C14415
C                                                                       014416
C                                                                       C14417
```

```
C
C        *****  DISCOS PROGRAM DATA STREAM  *****
C
C                     **************
C
C
C******************************************************************
C*
C* MAIN PROGRAM
C*
C******************************************************************
C
C          NIT = INPUT TAPE NUMBER
C
C--------------------- DENOTES WHERE DATA
C--------------------- IS READ INTO PROGRAM
C
C
C-----999 CALL START
C
C---------CALL COMENT
C
C---------CALL DYNS10
C
C---------CALL DYNS20
C
C
C              IFLNER = LINEARIZATION FLAG = IPDATA(3)
C              NOPLOT = PLOT CONTROL FLAG  = IPDATA(2)
C
C---------IF (IFLNER .EQ. 1) CALL DYNS40
C
C---------IF (NOPLOT .GT. 0) CALL DYNS30
C
C         GO TO 999
C
C         END
C
C
C
C
C
C******************************************************************
C*
C* SUBROUTINE START  - INPUT IDENTIFICATIONS
C*
C******************************************************************
C
C
C---------READ(NIT,FORMAT = A5,4X,3A6) IRUNNO, (UNAME(I),I=1,3)
C
C                 IRUNNO = RUN IDENTIFICATION NUMBER (5 CHARACTERS)
C                 UNAME  = USERS NAME (18 CHARACTERS)
C
C                 IRUNNO EQ 4HSTOP - TERMINATE THE RUN
C                 IRUNNO NE 4HSTOP - CONTINUE THE RUN
C
C                 DEBUG(I) = .TRUE.  IF DEBUG PRINT STATEMENTS CODED IN
C                                    SUBROUTINE DEBUG # I ARE TO BE USED
C                          = .FALSE. IF NO DEBUG PRINTING DESIRED IN
```

```
C                                   SUBROUTINE DEBUG # 1                      C14478
C                                                                             C14479
C             IF (IRUNNO.NE.ONDEBUG ) GO TO 1                                 014480
C                                                                             C14481
C--------------READ(NIT,FORMAT = 60L1) (DEBUG(I),I=1,120)                     C14482
C                                                                             C14483
C           1 CONTINUE                                                        C14484
C                                                                             014485
C                                                                             014486
C                                                                             C14487
C----------READ(NIT,FORMAT = 12A6) (TITLE1(I),I=1,12)                         014488
C----------READ(NIT,FORMAT = 12A6) (TITLE2(I),I=1,12)                         C14489
C                                                                             C14490
C                 TITLE1 = 72 CHARACTER USER SUPPLIED TITLE                   014491
C                 TITLE2 = 72 CHARACTER USER SUPPLIED TITLE                   C14492
C                                                                             C14493
C         RETURN                                                              014494
C         END                                                                 C14495
C                                                                             014496
C                                                                             014497
C                                                                             014498
C                                                                             014499
C                                                                             C14500
C************************************************************************     014501
C*                                                                            C14502
C* SUBROUTINE COMENT - INPUT USER SUPPLIED COMMENTS                           014503
C*                                                                            C14504
C************************************************************************     C14505
C                                                                             C14506
C                                                                             014507
C----------READ(NIT,FORMAT = 13A6,1X,A1) (ICOMRK(I),I=1,13), IPGHD            014508
C                                                                             C14509
C                 ICOMRK = 78 CHARACTER USER SUPPLIED COMMENT                 014510
C                 IPGHD  = NEW PAGE FLAG                                      C14511
C                                                                             C14512
C                      IPGHD = 1HP -- NEW PAGE BEFORE PRINTING                014513
C                      THERE IS NO LIMIT TO THE NUMBER OF COMMENT CARDS       014514
C                      BUT THE LAST COMMENT CARD MUST CONTAIN ZERO IN         C14515
C                      COLUMNS 1 THRU 13                                      C14516
C                                                                             C14517
C         RETURN                                                              C14518
C         END                                                                 014519
C                                                                             C14520
C                                                                             C14521
C                                                                             014522
C                                                                             C14523
C                                                                             C14524
C************************************************************************     C14525
C*                                                                            014526
C* SUBROUTINE [illegible] - INPUT PRIMARY DATA                                014527
C*                                                                            014528
C************************************************************************     C14529
C                                                                             014530
C                                                                             C14531
C------READ(NIT,FORMAT = 5I5) NS, NR, NSPT, [illegible], [illegible]          014532
C                                                                             C14533
C                NS     = NUMBER OF [illegible]             (NS.GE.2)         014534
C                NR     = NUMBER OF RINGS                                     C14535
C                NSPT   = NUMBER OF SENSOR POINTS           (NSPT.GE.1)       014536
C                [illegible] = NUMBER OF [illegible]                          014537
```

```
C                 NDELTA = NUMBER OF ADDITIONAL DIFFERENTIAL EQS
C                          REQUIRED - CONTAINS CONTROL SYSTEM
C                          VARIABLES
C
C
C----------CALL READIM (ITOPOL, 2, NH, 2, NHMAX)
C
C                  MATRIX SIZE 2 BY NH TO DESCRIBE TOPOLOGY
C
C                  FOR THE JTH HINGE -
C
C                  ITOPOL(1,J) = BODY N
C                  ITOPOL(2,J) = BODY M
C                    BODY N CONNECTED TO BODY M AT HINGE J
C                        ITOPOL(1,1) = 1 BY DEFINITION (BODY 1)
C                        ITOPOL(2,1) = 0 BY DEFINITION (INERTIAL REF)
C                     THE G-FRAME IS ON BODY N
C                     THE P-FRAME IS ON BODY M
C                     MOTION OF THE G-FRAME IS MEASURED RELATIVE
C                             TO THE P-FRAME
C
C
C----------CALL READIM (IRGFLX, 1, NB, 1, NBMAX)
C
C                  VECTOR SIZE 1 BY NB TO DEFINE NUMBER OF ELASTIC MODES
C
C                  FOR THE JTH BODY -
C
C                  IRGFLX(J) = 0  - RIGID BODY
C                  IRGFLX(J) = N  - FLEXIBLE BODY WITH N MODES
C
C
C----------CALL READIM (IFTSMW, 1, NSPT, 1, NSPMAX)
C
C                  VECTOR SIZE 1 BY NSPT TO DEFINE SENSOR POINT LOCATIONS
C
C                  FOR THE JTH SENSOR POINT -
C
C                  IFTSMW(J) = BODY NUMBER FOR SENSOR POINT J
C
C                  AT LEAST ONE SENSOR POINT MUST BE DEFINED IN
C                  THE SIMULATION MODEL
C
C
C
C
C
C               *** SUMMARY OF EULER ROTATION TYPES ***
C
C
C                       ITYPE    PERMUTATION ORDER
C
C                         1          (1,2,3)
C
C                         2          (1,2,1)
C
C                         3          (1,3,1)
C
C                         4          (1,3,2)
C
```

```
C                     5          (2,3,1)                                      014598
C                                                                             014599
C                     6          (2,3,2)                                      014600
C                                                                             014601
C                     7          (2,1,2)                                      014602
C                                                                             014603
C                     8          (2,1,3)                                      014604
C                                                                             014605
C                     9          (3,1,2)                                      014606
C                                                                             014607
C                    10          (3,1,3)                                      014608
C                                                                             014609
C                    11          (3,2,3)                                      014610
C                                                                             014611
C                    12          (3,2,1)                                      014612
C                                                                             014613
C                                                                             014614
C----------CALL READIT (IHDATA, 7, NH, 7, NHMAX)                               014615
C                                                                             014616
C              MATRIX SIZE 7 BY NH TO DEFINE HINGE POINT                      014617
C              CONSTRAINT DATA                                                014618
C                                                                             014619
C              FOR THE JTH HINGE -                                            014620
C                                                                             014621
C              IHDATA(1,J) = EULER ROTATION TYPE TO ORIENT Q-TRIAD            014622
C                                 WRT P-TRIAD AT HINGE J (ITYPE)              014623
C              IHDATA(2,J) = HINGE CONSTRAINT TYPE - THETA 1 ROTATION         014624
C              IHDATA(3,J) = HINGE CONSTRAINT TYPE - THETA 2 ROTATION         014625
C              IHDATA(4,J) = HINGE CONSTRAINT TYPE - THETA 3 ROTATION         014626
C              IHDATA(5,J) = HINGE CONSTRAINT TYPE - X TRANSLATION            014627
C              IHDATA(6,J) = HINGE CONSTRAINT TYPE - Y TRANSLATION            014628
C              IHDATA(7,J) = HINGE CONSTRAINT TYPE - Z TRANSLATION            014629
C                                                                             014630
C                 ROTATIONS REFER TO ORDER DEFINED BY ITYPE                   014631
C                                                                             014632
C                 TRANSLATIONS REFER TO P-TRIAD                               014633
C                                                                             014634
C                 IHDATA(2-7,J) = 0 - NO CONSTRAINT                           014635
C                 IHDATA(2-7,J) = 1 - FIXED CONSTRAINT                        014636
C                 IHDATA(2-7,J) = 2 - RHEONOMIC CONSTRAINT                    014637
C                                                                             014638
C                                                                             014639
C                                                                             014640
C                                                                             014641
C                                                                             014642
C                                                                             014643
C                                                                             014644
C              NOTE -- NUMBER OF BETA STATE VARIABLES COMPUTED FROM           014645
C                      IHDATA AS SUM OF NUMBER OF ZEROS + SUM OF              014646
C                      NUMBER OF TWOS IN ROWS 2 THRU 7                        014647
C                                                                             014648
C                      NUMBER OF CONSTRAINTS COMPUTED FROM                    014649
C                      IHDATA AS SUM OF NUMBER OF ONES + SUM OF               014650
C                      NUMBER OF TWOS IN ROWS 2 THRU 7                        014651
C                                                                             014652
C                                                                             014653
C----------CALL READ (BETAH, 6, NH, 6, NHMAX)                                  014654
C                                                                             014655
C              MATRIX SIZE 6 BY NH TO DEFINE INITIAL VALUES                   014656
C              OF BETA WHICH ORIENT TWO BODIES ASSOCIATED                     014657
```

```
C                 WITH EACH HINGE
C
C                 FOR THE JTH HINGE -
C
C                 BETAH(1,J) = THETA 1 ROTATION
C                 BETAH(2,J) = THETA 2 ROTATION
C                 BETAH(3,J) = THETA 3 ROTATION
C                 BETAH(4,J) = X TRANSLATION
C                 BETAH(5,J) = Y TRANSLATION
C                 BETAH(6,J) = Z TRANSLATION
C
C
C----------CALL READ (BETAHD, 6, NH, 6, NHMAX)
C
C                 MATRIX SIZE 6 BY NH TO DEFINE INITIAL VALUES
C                 OF BETA DOT - TIME DERIVATIVE OF BETAH
C                 DESCRIBED PREVIOUSLY
C
C                 FOR THE JTH HINGE -
C
C                 BETAHD(1,J) = THETA 1 ROTATION RATE
C                 BETAHD(2,J) = THETA 2 ROTATION RATE
C                 BETAHD(3,J) = THETA 3 ROTATION RATE
C                 BETAHD(4,J) = X TRANSLATION RATE
C                 BETAHD(5,J) = Y TRANSLATION RATE
C                 BETAHD(6,J) = Z TRANSLATION RATE
C
C
C                 NOTE -- IF THE CORRESPONDING CONSTRAINT TYPE
C                         IS 1 OR 2, THE INITIAL BETAHD(1-6,J),
C                         INPUT HERE, WILL BE IGNORED AND SET TO
C                         ZERO OR TO THE RHEONOMICALLY PRESCRIBED
C                         VALUE, RESPECTIVELY
C
C
C
C
C
C
C
C
C      NCFMC = NUMBER OF MOMENTUM WHEELS
C
C          IF (NCFMO .EQ. 0) GO TO 91
C
C
C----------CALL READIM (IMO, 3, NCFMO, 3, NMWMAX)
C
C                 MATRIX SIZE 3 BY NCFMO TO DEFINE BASIC
C                 MOMENTUM WHEEL DATA
C
C                 FOR THE JTH MOMENTUM WHEEL -
C
C                 IMO(1,J) = SENSOR POINT NUMBER FOR THE WHEEL
C                 IMO(2,J) = SPIN AXIS NUMBER FOR THE WHEEL (1,2 OR 3)
C                 IMO(3,J) = 1 ACTIVE WHEEL
C                          = 0 CONSTANT SPEED WHEEL
C
C
C----------CALL READ (AMO, 2, NCFMC, 2, NMWMAX)
```

```
C                                                                       C14718
C              MATRIX SIZE 2 BY NOEMG TO DEFINE MOMENTUM                C14719
C              WHEEL DATA                                               014720
C                                                                       C14721
C              FOR THE JTH MOMENTUM WHEEL -                             C14722
C                                                                       014723
C              AMW(1,J) = INITIAL WHEEL SPIN RATE                       014724
C              AMW(2,J) = SPIN INERTIA                                  014725
C                                                                       014726
C       41 CONTINUE                                                     C14727
C                                                                       014728
C                                                                       014729
C----------CALL READ (TMDATA, 1, 3, 1, 3)                               014730
C                                                                       014731
C              VECTOR SIZE 1 BY 3 CONTAINING TIME HISTORY               C14732
C              INTEGRATION CONTROLS                                     014733
C                                                                       014734
C              TMDATA(1) = INITIAL TIME      (START)                    014735
C              TMDATA(2) = TIME INTERVAL     (DELTAT)                   C14736
C              TMDATA(3) = TERMINATION TIME (LAST  )                    014737
C                                                                       C14738
C                                                                       C14739
C                                                                       C14740
C                                                                       014741
C                                                                       C14742
C                                                                       014743
C                                                                       C14744
C-----------CALL READ (IODATA, 1, 3, 1, 3)                              C14745
C                                                                       C14746
C              VECTOR SIZE 1 BY 3 CONTAINING OUTPUT                     C14747
C              CONTROL DATA                                             014748
C                                                                       014749
C              IODATA(1) = PRINT CONTROL FOR TIME RESPONSE - PRINT      C14750
C                          EVERY IODATA(1) MULTIPLES OF DELTAT          014751
C              IODATA(2) = PLOT CONTROL FOR TIME RESPONSE - SAVE        014752
C                          EVERY IODATA(2) MULTIPLES OF DELTAT          014753
C              IODATA(3) = 0 PERFORM NONLINEAR TIME RESPONSE            C14754
C                        = 1 PERFORM LINEAR TIME OR FREQUENCY           C14755
C                            RESPONSE                                   C14756
C                                                                       014757
C                                                                       C14758
C-----------CALL READ (CNTSTA, 1, NUSTAR, 1, NCCNT)                     014759
C                                                                       C14760
C              VECTOR SIZE 1 BY NUSTAR FOR CONTROL STATE VARIABLES      C14761
C              AND USER SUPPLIED VARIABLES                              C14762
C                                                                       014763
C              THIS IS A CATCH-ALL VECTOR FOR USER PAR DATA             C14764
C                                                                       014765
C              ITS LOCATION IS NOT FIXED BUT IS USER CONTROLLED AND IS  014766
C              IDENTIFIED BY USER SUPPLIED EQUIVALENCE MAP              014767
C              IN USER PAR                                              C14768
C                                                                       014769
C              FIRST [illegible] IN VECTOR MUST CONTAIN INITIAL VALUES  C14770
C              FOR CONTROL VARIABLES IN STATE VECTOR                    014771
C                                                                       C14772
C              ADDITIONAL SPACE IS AVAILABLE TO THE USER                C14773
C                                                                       014774
C                                                                       014775
C      ------CALL [illegible] (xxx, 1, [illegible], 1, [illegible])     C14776
                                                                        C14777
```

```
C                 VECTOR SIZE 1 BY 4 FOR GRAVITY GRADIENT DATA
C
C                 WV(1) = PROJECTION OF GRAVITY VECTOR ON X INERTIAL AXIS
C                 WV(2) = PROJECTION OF GRAVITY VECTOR ON Y INERTIAL AXIS
C                 WV(3) = PROJECTION OF GRAVITY VECTOR ON Z INERTIAL AXIS
C                 WV(4) = RADIUS VECTOR FROM GRAVITY SOURCE TO GENERAL
C                         VICINITY OF BODY CLUSTER
C
C
C                 NOTE -- IF(WV(1)**2 + WV(2)**2 + WV(3)**2)
C                                   EQ 0, WV(4) IS IGNORED
C                                   NE 0, WV(4) MUST BE GT 1
C
C                        (GRAVITY VECTOR COMPONENTS IN
C                        UNITS OF ACCELERATION)
C
C
C
C
C
C
C
C      NB = NUMBER OF BODIES
C
C          DO 20 N=1,NB
C
C                 IHGFLX(N) EQ 0 -  RIGID BODY
C                 IHGFLX(N) GT 0 -  FLEXIBLE BODY
C
C          IF (IHGFLX(N) .EQ. 0) GO TO 25
C
C----------READ(NIT,FORMAT = I5) NTYPE
C
C                 NTYPE = 1 - LUMPED MASS REPRESENTATION
C                 NTYPE = 2 - CONSISTENT MASS REPRESENTATION
C
C          IF (NTYPE .EQ. 1) CALL RSMCOL(N)    --- SEE FOLLOWING
C
C          IF (NTYPE .EQ. 2) CALL RSMCOC(N)    --- SEE FOLLOWING
C
C          GO TO 20
C
C       25 CALL RRIGID(N)    --- SEE FOLLOWING
C
C       20 CONTINUE
C
C          RETURN
C          END
C
C
C
C
C
C****************************************************************
C*
C* SUBROUTINE RRIGID - INPUT FOR RIGID BODY
C*
C****************************************************************
C
C
```

```
C-----------READ(NIT,FORMAT = [illegible]) (V(J),J=1,3)                     014838
C                                                                           014839
C              [illegible]                                                  014840
C                                                                           014841
C              V(J) [illegible]                                             014842
C              V(1) = X [illegible] BY [illegible] C.G. [illegible] TRIAD)  014843
C              V(2) = Y [illegible] BY [illegible] C.G. [illegible] TRIAD)  014844
C              V(3) = Z [illegible] BY [illegible] C.G. [illegible] TRIAD)  014845
C                                                                           014846
C                                                                           014847
C-----------[illegible](V(J),J=1,[illegible])                               014848
C                                                                           014849
C              [illegible]                                                  014850
C                                                                           014851
C              V(1) = JXX                                                   014852
C              V(2) = JYY                                                   014853
C              V(3) = JZZ [illegible] (IN [illegible] - BODY TRIAD))        014854
C              V(4) = JXY    [illegible] POINT, NOT BODY C.G)               014855
C              V(5) = JXZ                                                   014856
C              V(6) = JYZ                                                   014857
C                                                                           014858
C                                        JXX -JXY -JXZ                      014859
C                INERTIA MATRIX =       -JXY  JYY -JYZ                      014860
C                                       -JXZ -JYZ  JZZ                      014861
C              [illegible] -- J(XY) = [illegible] (XY DM), ETC.             014862
C                                                                           014863
C          [illegible] HINGE 1.                                             014864
C              BODY 1                                                       014865
C                                                                           014866
C              HINGE POINT 1 IS AT BODY 1 REFERENCE POINT                   014867
C                                                                           014868
C          [illegible] I=1,NH                                               014869
C                                                                           014870
C                                                                           014871
C-----------READ(NIT,FORMAT = [illegible]) NUM, ITYPE                       014872
C                                                                           014873
C              NUM   = HINGE NUMBER                                         014874
C              ITYPE = EULER ROTATION TYPE TO ORIENT HINGE                  014875
C                      TRIAD WRT BODY TRIAD                                 014876
C                                                                           014877
C                                                                           014878
C                                                                           014879
C                                                                           014880
C                                                                           014881
C                                                                           014882
C                                                                           014883
C-----------READ(NIT,FORMAT = 3G10.3) (V(J),J=1,3)                          014884
C                                                                           014885
C              EULER ANGLES TO ORIENT HINGE TRIAD - PERMUTATION             014886
C              ORDER DEFINED BY ITYPE                                       014887
C                                                                           014888
C              V(1) = THETA 1 (FIRST ROTATION)                              014889
C              V(2) = THETA 2 (SECOND ROTATION)                             014890
C              V(3) = THETA 3 (THIRD ROTATION)                              014891
C                                                                           014892
C                                                                           014893
C-----------READ(NIT,FORMAT = 3G10.3) (CH(J),J=1,3)                         014894
C                                                                           014895
C              VECTOR TO POSITION HINGE TRIAD WRT BODY TRIAD                014896
C                                                                           014897
```

```
C                    DH(1) = X (BODY REF POINT TO HINGE POINT, BODY TRIAD)
C                    DH(2) = Y (BODY REF POINT TO HINGE POINT, BODY TRIAD)
C                    DH(3) = Z (BODY REF POINT TO HINGE POINT, BODY TRIAD)
C
C      10 CONTINUE
C
C         NSB = NUMBER OF SENSOR POINTS ON BODY K
C
C         IF (NSB .LE. 0) RETURN
C
C         DO 20 I=1,NSB
C
C
C----------READ(NIT,FORMAT = 2I5) NOS, ITYPE
C
C                    NOS   = SENSOR POINT NUMBER
C                    ITYPE = EULER ROTATION TYPE TO ORIENT SENSOR POINT
C                            TRIAD WRT BODY TRIAD
C
C
C----------READ(NIT,FORMAT = 3D10.3) (V(J),J=1,3)
C
C                    EULER ANGLES TO ORIENT SENSOR POINT TRIAD - PERMUTATION
C                    ORDER DEFINED BY ITYPE
C
C                    V(1) = THETA 1 (FIRST ROTATION)
C                    V(2) = THETA 2 (SECND ROTATION)
C                    V(3) = THETA 3 (THIRD ROTATION)
C
C
C
C
C
C
C
C
C----------READ(NIT,FORMAT = 3D10.3) (DS(J),J=1,3)
C
C                    VECTOR TO POSITION SENSOR POINT TRIAD WRT BODY TRIAD
C
C                    DS(1) = X (BODY REF POINT TO SENSOR POINT, BODY TRIAD)
C                    DS(2) = Y (BODY REF POINT TO SENSOR POINT, BODY TRIAC)
C                    DS(3) = Z (BODY REF POINT TO SENSOR POINT, BODY TRIAC)
C
C      20 CONTINUE
C
C         RETURN
C         END
C
C
C
C
C
C*************************************************************
C*
C* SUBROUTINE MSMODE - INPUT FOR FLEXIBLE BODY, LUMPED MASS MATRIX
C*
C*************************************************************
C
C
```

```
C----------CALL READ (A, NJ, 1, KJOINT, KMODE)
C
C               MATRIX SIZE NJ BY 1 WHERE NJ = NUMBER OF JOINTS
C               ON BODY N
C
C               FOR THE ITH JOINT -
C
C               A(I,1) = JOINT MASS
C
C
C----------CALL READ (A, NJ, 6, KJOINT, KMODE)
C
C               MATRIX SIZE NJ BY 6
C
C               FOR THE ITH JOINT -
C
C               A(I,1) = JOINT INERTIA, JXX
C               A(I,2) = JOINT INERTIA, JYY
C               A(I,3) = JOINT INERTIA, JZZ
C               A(I,4) = JOINT INERTIA, JXY
C               A(I,5) = JOINT INERTIA, JXZ
C               A(I,6) = JOINT INERTIA, JYZ
C
C                                          JXX -JXY -JXZ
C                    INERTIA MATRIX = -JXY  JYY -JYZ
C                                         -JXZ -JYZ  JZZ
C               NOTE -- J(XY) = VOL INTEGRAL OF (XY DM), ETC.
C
C
C          [illegible]
C
C
C----------CALL READ (A, NJ, 3, KJOINT, KMODE)
C
C               MATRIX SIZE NJ BY 3
C
C               FOR THE ITH JOINT -
C
C             K=1   A(I,1) = JOINT STATIC MASS MOMENT, SX
C                   A(I,2) = JOINT STATIC MASS MOMENT, SY
C                   A(I,3) = JOINT STATIC MASS MOMENT, SZ
C
C             K=2   A(I,1) = X (BODY REF POINT TO JOINT, BODY TRIAD)
C                   A(I,2) = Y (BODY REF POINT TO JOINT, BODY TRIAD)
C                   A(I,3) = Z (BODY REF POINT TO JOINT, BODY TRIAD)
C
C          IN ADDITION
C
C
C                                       M     0     0     0    SZ   -SY
C                                       0     M     0   -SZ     0    SX
C                                       0     0     M    SY   -SX     0
C          JOINT INERTIA MATRIX =       0   -SZ    SY   JXX  -JXY  -JXZ
C                                      SZ     0   -SX  -JXY   JYY  -JYZ
C                                     -SY    SX     0  -JXZ  -JYZ   JZZ
C
C
C
C
```

```
C
C
C          DO 10 K=1,6
C
C
C----------CALL READ (A, NJ, NE, KJOINT, KMODE)
C
C                   MATRIX SIZE NJ BY NE WHERE NE = NUMBER OF ELASTIC
C                   MODES RETAINED FOR BODY N
C
C                   FOR THE ITH JOINT -
C
C               K=1  A(I,J) = X DISPLACEMENT AT JOINT, MODE J
C               K=2  A(I,J) = Y DISPLACEMENT AT JOINT, MODE J
C               K=3  A(I,J) = Z DISPLACEMENT AT JOINT, MODE J
C               K=4  A(I,J) = THETA X ROTATION AT JOINT, MODE J
C               K=5  A(I,J) = THETA Y ROTATION AT JOINT, MODE J
C               K=6  A(I,J) = THETA Z ROTATION AT JOINT, MODE J
C
C       10 CONTINUE
C
C
C          DO 20 K=1,2
C
C
C----------CALL READ (A, NE, NE, KJOINT, KMODE)
C
C                   MATRIX SIZE NE BY NE
C
C               K=1  A = MODAL STIFFNESS
C
C               K=2  A = MODAL DAMPING
C
C       20 CONTINUE
C
C
C----------READ(NIT,FORMAT = 5D10.5) (A(J),J=1,NE)
C
C                   VECTOR OF INITIAL MODAL DEFLECTION COORDINATES
C
C----------READ(NIT,FORMAT = 5D10.5) (A(J),J=1,NE)
C
C                   VECTOR OF INITIAL MODAL VELOCITY COORDINATES
C
C
C
C
C
C
C
C
C                   NOTE -- FOLLOWING EULER ANGLES MEASURED
C                          IN UNDEFORMED CONFIGURATION
C
C
C          NHG = NUMBER OF HINGES ON BODY N - EXCLUSIVE OF HINGE 1,
C                BODY 1
C
C          DO 110 L=1,NHG
C
```

```
C
C ----READ(INIT,FORMAT = (3I5)) NSP, ITYPE, JOINT
C
C         NSP   = SENSOR POINT NUMBER
C         ITYPE = EULER ROTATION TYPE TO ORIENT SENSOR
C                 TRIAD [illegible] BY TRIAD
C         JOINT = JOINT NUMBER CORRESPONDING TO SENSOR POINT
C
C
C ----READ(INIT,FORMAT = (3D10.3)) (AV(J),J=1,3)
C
C         EULER ANGLES TO ORIENT SENSOR TRIAD - PERMUTATION
C         ORDER DEFINED BY ITYPE
C
C         AV(1) = THETA 1 (FIRST ROTATION)
C         AV(2) = THETA 2 (SECOND ROTATION)
C         AV(3) = THETA 3 (THIRD ROTATION)
C
C    1[illegible] CONTINUE
C
C
C      [illegible] SENSOR POINTS [illegible]
C
C      IF (NSP .EQ. 0) RETURN
C
C      DO 1[illegible] I=1,NSP
C
C
C-----------READ(INIT,FORMAT = (3I5)) NSP, ITYPE, JOINT
C
C         NSP   = SENSOR POINT NUMBER
C         ITYPE = EULER ROTATION TYPE TO ORIENT SENSOR POINT
C                 TRIAD [illegible] BY TRIAD
C         JOINT = JOINT NUMBER CORRESPONDING TO SENSOR POINT
C
C
C
C
C
C
C
C
C-----------READ(INIT,FORMAT = (3D10.3)) (AV(J),J=1,3)
C
C         EULER ANGLES TO ORIENT SENSOR POINT TRIAD - PERMUTATION
C         ORDER DEFINED BY ITYPE
C
C         AV(1) = THETA 1 (FIRST ROTATION)
C         AV(2) = THETA 2 (SECOND ROTATION)
C         AV(3) = THETA 3 (THIRD ROTATION)
C
C    1[illegible] CONTINUE
C
C      RETURN
C      END
C
C
C
C
C
```

```
C********************************************************************************   015138
C*                                                                                  015139
C*  SUBROUTINE MSRIGD - INPUT FOR FLEXIBLE BODY, CONSISTENT MASS MATRIX             015140
C*                                                                                  015141
C********************************************************************************   015142
C                                                                                   015143
C                                                                                   015144
C----------CALLING FORMAT = RIGD (IFORM, IDIAK, IDIAD)                              015145
C                                                                                   015146
C                         IFORM = RIGID BODY MODE CONTROL VARIABLE                  015147
C                                                                                   015148
C                               = 0  WHEN RIGID BODY MODES CAN BE                   015149
C                                    OPERATED ON VIA ROW-COL                        015150
C                                    INTERCHANGES TO OBTAIN A RIGID                 015151
C                                    BODY MODAL MATRIX OF THE FORM-                 015152
C                                                                                   015153
C                            **  **   **                    ** **    **             015154
C                            * HX *   * 1  0  0  0  Z -Y  * * QX   *             015155
C                            *    *   *                   * *      *             015156
C                            * HY *   * 0  1  0 -Z  0  X  * * QY   *             015157
C                            *    *   *                   * *      *             015158
C                            * HZ *   * 0  0  1  Y -X  0  * * QZ   *             015159
C                            *    * = *                   * *      *             015160
C                            * TX *   * 0  0  0  1  0  0  * * QTX  *             015161
C                            *    *   *                   * *      *             015162
C                            * TY *   * 0  0  0  0  1  0  * * QTY  *             015163
C                            *    *   *                   * *      *             015164
C                            * TZ *   * 0  0  0  0  0  1  * * QTZ  *             015165
C                            **  **   **                    ** **    **             015166
C                                                                                   015167
C                                                                                   015168
C                               = 1  WHEN ABOVE IS NOT TRUE, SUCH                   015169
C                                    AS FOR AN ORTHONORMAL SET OF                   015170
C                                    RIGID BODY VECTORS, DATA WILL                  015171
C                                    BE READ IN LATER AND THE ABOVE                 015172
C                                    FORM WILL BE CREATED WITHIN                    015173
C                                    THIS SUBROUTINE                                015174
C                                                                                   015175
C                                                                                   015176
C                                                                                   015177
C                                                                                   015178
C                                                                                   015179
C                                                                                   015180
C                                                                                   015181
C                         IDIAK = 0  IF MODAL STIFFNESS MATRIX                      015182
C                                    IS NOT DIAGONAL                                015183
C                                                                                   015184
C                               = 1  IF MODAL STIFFNESS MATRIX                      015185
C                                    IS DIAGONAL                                    015186
C                                                                                   015187
C                                                                                   015188
C                         IDIAD = 0  IF MODAL DAMPING MATRIX                        015189
C                                    IS NOT DIAGONAL                                015190
C                                                                                   015191
C                               = 1  IF MODAL DAMPING MATRIX                        015192
C                                    IS DIAGONAL                                    015193
C                                                                                   015194
C                                                                                   015195
C----------CALL READIN (JDOF, NA, NO, KJDOF, 6)                                     015196
C                                                                                   015197
```

```
C                 MATRIX SIZE NA BY 6 DEFINING JOINT DEGREES                   015198
C                 OF FREEDOM FOR NA JOINTS ON BODY N                           C15199
C                                                                              015200
C                 FOR THE ITH JOINT -                                          C15201
C                                                                              015202
C                  JDOF(I,1) = X DEG. OF FREEDOM FOR POINT I.                  C15203
C                  JDOF(I,2) = Y DEG. OF FREEDOM FOR POINT I.                  015204
C                  JDOF(I,3) = Z DEG. OF FREEDOM FOR POINT I.                  015205
C                  JDOF(I,4) =TX DEG. OF FREEDOM FOR POINT I.                  C15206
C                  JDOF(I,5) =TY DEG. OF FREEDOM FOR POINT I.                  C15207
C                  JDOF(I,6) =TZ DEG. OF FREEDOM FOR POINT I.                  C15208
C                                                                              C15209
C                        POINT I IS CONSISTANT WITH THE                        015210
C                        MODAL MATRIX ROW I, BEFORE REORDING.                  015211
C                                                                              015212
C                        JDOF WILL BE INTERROGATED AND THE                     015213
C                        COORDINATES WILL BE ORDERED AS --                     015214
C                                                                              C15215
C                         HX(I) FOR JOINTS 1 THRU NA                           015216
C                         HY(I) FOR JOINTS 1 THRU NA                           015217
C                         HZ(I) FOR JOINTS 1 THRU NA                           C15218
C                         TX(I) FOR JOINTS 1 THRU NA                           015219
C                         TY(I) FOR JOINTS 1 THRU NA                           015220
C                         TZ(I) FOR JOINTS 1 THRU NA                           015221
C                                                                              015222
C                                                                              C15223
C                                                                              015224
C                                                                              015225
C                                                                              C15226
C                                                                              015227
C                                                                              C15228
C-----------CALL [illegible] (JV, I, NROOT, I, KAJ)                             015229
C                                                                              C15230
C                 VECTOR SIZE 1 BY NROOT = NUMBER OF RIGID BODY MODES +        015231
C                                          NUMBER OF ELASTIC MODES             015232
C                                                                              015233
C                 JV(J) = IND   WHERE IND IS COLUMN NO. WHICH COL(J) OF        C15234
C                               ORIGINAL MODAL MATRIX WILL APPEAR IN           015235
C                               REVISED MODE MATRIX                            015236
C                     IND GT 0 - REPLACE COLUMN                                015237
C                     IND EQ 0 - DELETE COLUMN                                 015238
C                     IND LT 0 - REPLACE COLUMN, CHANGE SIGNS                  015239
C                                                                              C15240
C                                                                              015241
C-----------CALL [illegible] (A, 6NA, [illegible], NAJ, KAJ)                    015242
C                                                                              C15243
C                 MATRIX SIZE 6NA BY 6NA CONTAINS CONSISTENT                   015244
C                 MASS REPRESENTATION.                                         015245
C                                                                              C15246
C                 THE COLUMN COORDINATE ORDER MUST BE CONSISTENT               015247
C                 WITH THE DEGREE OF FREEDOM TABLE, JDOF                       015248
C                                                                              C15249
C                                                                              015250
C-------CALL [illegible] (A, 6NA, NROOT, KA, KAJ)                               015251
C                                                                              C15252
C                 MATRIX SIZE 6NA BY NROOT CONTAINS MODAL DEFINITION           015253
C                                                                              C15254
C                 THE COLUMNS (MODE NUMBER) WILL BE REORDERED BY               C15255
C                 THE INPUT VECTOR JV                                          015256
C                                                                              C15257
```

```
C
C          IF (IDIAK .EQ. 0  .AND.  IDIAD .EQ. 0) GO TO 11
C
C----------CALL READ (OM2, 1, NMODT, 1, KA3)
C
C                    VECTOR SIZE 1 BY NMODT CONTAINING SQUARES OF NATURAL
C                    FREQUENCIES
C
C                    OM2(J) = SQUARE OF JTH NATURAL FREQUENCY CORRESPONDING
C                    TO JTH INPUT MODE SHAPE
C
C      11 CONTINUE
C
C
C
C
C
C
C
C          IF (IFROM .EQ. 0) GO TO 5
C
C
C----------READ(NIT,FORMAT = I5) JTYPCL
C
C                    JTYPCL - REFERENCE JOINT NUMBER WHOSE GEOMETRIC
C                             POSITION COORDINATES WILL BE USED TO
C                             ESTABLISH RIGID BODY MODAL MATRIX
C
C----------READ(NIT,FORMAT = 3D10.3) (OM2(J),J=1,3)
C
C                    OM2(1) - X COMPONENT OF VECTOR THAT LOCATES JTYPCL PT
C                    OM2(2) - Y COMPONENT OF VECTOR THAT LOCATES JTYPCL PT
C                    OM2(3) - Z COMPONENT OF VECTOR THAT LOCATES JTYPCL PT
C
C                    VECTOR COMPONENTS REFERED TO BODY TRIAD
C
C       5 CONTINUE
C
C
C          IF (IDIAK .EQ. 1) GO TO 50
C
C----------CALL READ (A, NRA, NCA, KAD, KAB)
C
C                    STIFFNESS MATRIX -- SEE NOTE BELOW
C
C      50 CONTINUE
C
C          IF (IDIAD .EQ. 1) GO TO 60
C
C
C----------CALL READ (A, NRA, NCA, KAD, KAD)
C
C                    DAMPING MATRIX -- SEE NOTE BELOW
C
C          GO TO 61
C
C      60 CONTINUE
C
C
C----------READ(NIT,FORMAT = 6D10.3) (OM2(J),J=1,6)
```

```
C                                                                       015318
C          [illegible] ELASTIC MODES                                    015319
C          [illegible]                                                  015320
                                                                        015321
C          [illegible] ELASTIC MODE                                     015322
C                                                                       015323
      [illegible]                                                       015324
                                                                        015325
C                                                                       015326
C                                                                       015327
C                                                                       015328
C                                                                       015329
C                                                                       015330
C                                                                       015331
C              [illegible]                                              015332
C                                                                       015333
C                [illegible] WITH                                       015334
C                  [illegible] POSITION                                 015335
C                  [illegible] (I)                                      015336
C                  [illegible] AND THE FOLLOWING                        015337
C                  [illegible]                                          015338
C                                                                       015339
C                                                                       015340
C                        X(I),I=1,.......                               015341
C                        .                                              015342
C                        .                                              015343
C                        Y(I),I=1,.......                               015344
C                        .                                              015345
C                        .                                              015346
C                        Z(I),I=1,.......                               015347
C                        .                                              015348
C                        .                                              015349
C                        TX(I),I=1,.......                              015350
C                        .                                              015351
C                        .                                              015352
C                        TY(I),I=1,.......                              015353
C                        .                                              015354
C                        .                                              015355
C                        TZ(I),I=1,.......                              015356
C                        .                                              015357
C                        .                                              015358
C                                                                       015359
C                                                                       015360
C---------- READ(5,[illegible]), FORMAT = [illegible] (CM[illegible](J),J=1,NC)   015361
                                                                        015362
C               VECTOR SIZE 1 BY NC CONTAINING INITIAL                  015363
C               [illegible] COORDINATES                                 015364
C                                                                       015365
C                                                                       015366
C---------- READ(5,[illegible]), FORMAT = [illegible] (CM[illegible](J),J=1,NC)   015367
C                                                                       015368
C               VECTOR SIZE 1 BY NC CONTAINING INITIAL                  015369
C               MODAL VELOCITY COORDINATES                              015370
C                                                                       015371
C                                                                       015372
C                                                                       015373
C                                                                       015374
C                                                                       015375
C                                                                       015376
                                                                        015377
```

```
C                NOTE -- FOLLOWING EULER ANGLES MEASURED                  015378
C                        IN UNDEFORMED CONFIGURATION                      015379
C                                                                         015380
C                                                                         015381
C        NHB = NUMBER OF HINGES ON BODY N - EXCLUSIVE OF HINGE 1,         015382
C              BODY 1                                                     015383
C                                                                         015384
C        DO 110 L=1,NHB                                                   015385
C                                                                         015386
C                                                                         015387
C---------READ(NIT,FORMAT = 3I5) NGH, ITYPE, JOINT                        015388
C                                                                         015389
C                NGH   = HINGE NUMBER                                     015390
C                ITYPE = EULER ROTATION TYPE TO ORIENT HINGE              015391
C                        TRIAD WRT BODY TRIAD                             015392
C                JOINT = JOINT NUMBER CORRESPONDING TO HINGE POINT        015393
C                                                                         015394
C                                                                         015395
C---------READ(NIT,FORMAT = 3D10.3) (GM2(J),J=1,3)                        015396
C                                                                         015397
C                EULER ANGLES TO ORIENT HINGE TRIAD - PERMUTATION         015398
C                ORDER DEFINED BY ITYPE                                   015399
C                                                                         015400
C                GM2(1) = THETA 1 (FIRST ROTATION)                        015401
C                GM2(2) = THETA 2 (SECND ROTATION)                        015402
C                GM2(3) = THETA 3 (THIRD ROTATION)                        015403
C                                                                         015404
C     110 CONTINUE                                                        015405
C                                                                         015406
C                                                                         015407
C        NSB = NUMBER OF SENSOR POINTS ON BODY N                          015408
C                                                                         015409
C        IF (NSB .EQ. 0) RETURN                                           015410
C                                                                         015411
C        DO 120 L=1,NSB                                                   015412
C                                                                         015413
C                                                                         015414
C---------READ(NIT,FORMAT = 3I5) NGS, ITYPE, JOINT                        015415
C                                                                         015416
C                NGS   = SENSOR POINT NUMBER                              015417
C                ITYPE = EULER ROTATION TYPE TO ORIENT SENSOR POINT       015418
C                        TRIAD WRT BODY TRIAD                             015419
C                JOINT = JOINT NUMBER CORRESPONDING TO SENSOR POINT       015420
C                                                                         015421
C                                                                         015422
C                                                                         015423
C                                                                         015424
C                                                                         015425
C                                                                         015426
C                                                                         015427
C                                                                         015428
C                                                                         015429
C---------READ(NIT,FORMAT = 3D10.3) (GM2(J),J=1,3)                        015430
C                                                                         015431
C                EULER ANGLES TO ORIENT SENSOR POINT TRIAD - PERMUTATION  015432
C                ORDER DEFINED BY ITYPE                                   015433
C                                                                         015434
C                GM2(1) = THETA 1 (FIRST ROTATION)                        015435
C                GM2(2) = THETA 2 (SECND ROTATION)                        015436
C                GM2(3) = THETA 3 (THIRD ROTATION)                        015437
```

```
C
C     120 CONTINUE
C
C         RETURN
C         END
C
C
C
C
C
C*********************************************************************
C*
C* SUBROUTINE LYNS20 - INPUT USER PAK DATA (IF ANY)
C*
C*********************************************************************
C
C
C                 THE ONLY DATA READ HERE ARE THOSE ASSOCIATED
C                 WITH POLYNOMIAL DEFINITION OF TRANSFER
C                 FUNCTIONS
C
C                 DATA (IF ANY) IS READ ON THE FIRST CALL TO
C                 SUBROUTINE CONTRL WHERE THE INTEGER VARIABLE
C                 NPLY IS DEFINED BY A USER SUPPLIED DATA
C                 STATEMENT
C
C                 NPLY = NUMBER OF SETS OF POLYNOMIAL TRANSFER
C                        FUNCTION COEFFICIENTS TO READ
C
C         IF (NPLY .EQ. 0) GO TO 10
C
C         DO 20 K=1,NPLY
C
C         KL = 2*K-1
C
C
C----------CALL READ (CPLY(1,KL), NPLY(K), 2, [illegible], [illegible])
C
C      20 CONTINUE
C
C                 MATRIX SIZE NPLY(K) BY 2 OF POLYNOMIAL COEFFICIENTS
C
C                 CPLY(I,KL) = DENOMINATOR COEFFICIENTS IN
C                              DESCENDING ORDER
C                 CPLY(I,KL+1) = NUMERATOR COEFFICIENTS IN
C                              DESCENDING ORDER
C
C                 NOTE -- NPLY(K) IS DEFINED AT FIRST SUBROUTINE READ)
C                        AND IS ONE GREATER THAN THE ORDER OF THE
C                        DENOMINATOR POLYNOMIAL
C
C                        ORDER OF NUMERATOR POLYNOMIAL MUST NOT EXCEED
C                        ORDER OF DENOMINATOR POLYNOMIAL
C
C      10 CONTINUE
C
C         RETURN
C         END
C
C
```

```
C                                                                        015498
C                                                                        015499
C                                                                        015500
C*********************************************************************** 015501
C*                                                                       015502
C* SUBROUTINE DYNSSC - INPUT FOR TIME HISTORY PLOT OUTPUT                015503
C*                                                                       015504
C*********************************************************************** 015505
C                                                                        015506
C                                                                        015507
C----------READ(NIT,FORMAT = 10A8) (ICTITL(I),I=1,10)                   015508
C                                                                        015509
C                 80 CHARACTER MASTER PLOT TITLE WILL APPEAR             015510
C                 ON ALL FRAMES                                          015511
C                                                                        015512
C                                                                        015513
C----------READ(NIT,FORMAT = I5) NSET                                   015514
C                                                                        015515
C                             NSET = NO. OF PLOT SETS TO BE              015516
C                                    CYCLED THROUGH. EACH SET            015517
C                                    IS LIMITED TO 10 SEPARATE           015518
C                                    VARIABLES TO BE SELECTED            015519
C                                    FROM THE SET OF VARIABLES           015520
C                                    WRITTEN BY SUBROUTINE PLOTWR        015521
C                                    OR LPLTWR.                          015522
C                                                                        015523
C                                                                        015524
C         DO 1000 ISET=1,NSET                                            015525
C                                                                        015526
C                                                                        015527
C----------READ(NIT,FORMAT = I5) NPL                                    015528
C                                                                        015529
C                             NPL = NO. OF VARIABLES TO BE               015530
C                                   SELECTED FROM THE NPPLOT             015531
C                                   VARIABLES PREVIOUSLY WRITTEN         015532
C                                   ON TAPE 8 FOR PLOTTING.              015533
C                                  (NPL .LE. 10) FOR A SET               015534
C                                                                        015535
C                             NPPLOT = 10+2*NEQ+NEAR+NQ+NR+NQ+NR+NQ+NR+NO 015536
C                                      (NONLINEAR ANALYSIS)              015537
C                                                                        015538
C                                                                        015539
C                             NPPLOT = 10+2*NEQ                          015540
C                                      (LINEAR ANALYSIS)                 015541
C                                                                        015542
C                                WHERE ----                              015543
C                                                                        015544
C                                  NEQ = NO. OF EQUATIONS                015545
C                                        INTEGRATED.                     015546
C                                                                        015547
C                                 NEAR = NO. OF INFLOW VARIABLES         015548
C                                                                        015549
C                                   NQ = NO. OF Q VARIABLES              015550
C                                                                        015551
C                                   NR = 6*NO                            015552
C                                                                        015553
C                                   NO = NO. OF NODES.                   015554
C                                                                        015555
C                                                                        015556
C                                                                        015557
```

```
C                                                                      015558
C                                                                      015559
C                                                                      015560
C                                                                      015561
C                                                                      015562
C      [illegible]      [illegible]                                    015563
C                  *****                                               015564
C                  *                                                   015565
C      [illegible] *  [illegible]IDATA(J) * [illegible] A * [illegible]TA  015566
C                  *                                                   015567
C                  ****                                                015568
C                  [illegible]                                         015569
C                                                                      015570
C                                                                      015571
C                                                                      015572
C                                                                      015573
C                  [illegible]         [illegible]                     015574
C                  *****               *****                           015575
C                  *                   *                               015576
C      [illegible] *  [illegible](J) *  * [illegible](K) * [illegible] 015577
C                  *                   *                               015578
C                  *****               *****                           015579
C                  J=1                 K=1                             015580
C                                                                      015581
C                                                                      015582
C                                                                      015583
C                                                                      015584
C           [illegible]TA = [illegible] NUMBER OF [illegible] * [illegible] OF  015585
C                  [illegible] 2 THRU 7                                015586
C                  OF ARRAY IDATA                                      015587
C                                                                      015588
C                                                                      015589
C                                                                      015590
C                                                                      015591
C           [illegible] = [illegible] NUMBER OF [illegible] * SUM OF   015592
C                  [illegible] IN ROWS 2 THRU 7                        015593
C                  OF ARRAY IDATA                                      015594
C                                                                      015595
C                                                                      015596
C                                                                      015597
C                                                                      015598
C                                                                      015599
C                                                                      015600
C                                                                      015601
C                                                                      015602
C                                                                      015603
C              ORDER OF VARIABLES AND                                  015604
C              SIZE FOR A SINGLE RECORD                                015605
C              FROM A SINGLE TIME ID.                                  015606
C                                                                      015607
C                                                                      015608
C               VARIABLE ID.          SIZE                             015609
C                                                                      015610
C     NONLINEAR    TIME                 1                              015611
C     ANALYSIS                                                         015612
C                   Y                  NEQ                             015613
C                                                                      015614
C                  YDOT                NEQ                             015615
C                                                                      015616
C                 LAMBDA               [illegible]                     015617
```

```
C                                                                       015618
C                              3                ND                      015619
C                                                                       015620
C                           HX, HY, HZ.                                 015621
C                           HX, HY, HZ        3*NB                      015622
C                                                                       015623
C                           TOTAL ANGULAR                               015624
C                           MOMENTUM VECTOR                             015625
C                           COMPONENTS (X,Y,Z). 3                       015626
C                                                      INERTIAL         015627
C                           TOTAL LINEAR               FRAME.           015628
C                           MOMENTUM VECTOR                             015629
C                           COMPONENTS (X,Y,Z). 3                       015630
C                                                                       015631
C                           BODY KINETIC                                015632
C                           ENERGIES.           NB                      015633
C                                                                       015634
C                           BODY POTENTIAL                              015635
C                           ENERGIES.           NB                      015636
C                                                                       015637
C                           TOTAL ANGULAR MOM..                         015638
C                           TOTAL LINEAR MOM..                          015639
C                           TOTAL K.E..                                 015640
C                           TOTAL P.E..                                 015641
C                           TOTAL ENERGY        9                       015642
C                                                                       015643
C                                                                       015644
C                                                                       015645
C             LINEAR                                                    015646
C             ANALYSIS      TIME                1                       015647
C                                                                       015648
C                           Y                   NCD                     015649
C                           YDOT                NCD                     015650
C                                                                       015651
C                                                                       015652
C                                                                       015653
C                                                                       015654
C                                                                       015655
C                                                                       015656
C                                                                       015657
C----------READ(NIT,FORMAT = 10I5) (JVPL(J),J=1,NPL)                    015658
C                                                                       015659
C                                                                       015660
C                         JVPL(J) = INTEGER DEFINING GLOBAL             015661
C                                   LOCATION OF JTH SELECTED            015662
C                                   VARIABLE FROM THE NCPLOT            015663
C                                   LONG ARRAY.                         015664
C                                                                       015665
C     20 CONTINUE                                                       015666
C                                                                       015667
C                                                                       015668
C----------READ(NIT,FORMAT = 5I5) NCI, (NCD(I),I=1,3), NGRID            015669
C                                                                       015670
C                         NCI = ELEMENT LOCATION (LOCAL WRT             015671
C                               JVPL ARRAY) FOR THE INDEPEN-            015672
C                               DENT PLOT VARIABLE.                     015673
C                                                                       015674
C                         NCD = ELEMENT LOCATION (LOCAL WRT             015675
C                               JVPL ARRAY) FOR UP TO 3                 015676
C                               DEPENDENT VARIABLES TO PLOT             015677
```

```
C                                SIMULTANEOUSLY VERSES THE
C                                NCI DEPENDENT VARIABLE.
C
C                          NGRID = NO. OF PLOT FRAMES TO USE
C                                  FOR PLOTTING THE NCI-NCD
C                                  GROUP. IE. THE NO. OF FRAMES
C                                  TO USE SIDE BY SIDE TO
C                                  EXHAUST THE RANGE OF THE
C                                  INDEPENDENT VARIABLE.
C
C
C         IF (NCI .EQ. 0) GO TO 1000
C
C              THIS IS DUE TO PROCEDE TO NEXT SET
C
C              THIS OPEN ENDED LOOP PERMITS MANY SELECTIONS OF THE
C              SAME DATA VARIABLES WITH REGARD TO INDEPENDENT AND
C              DEPENDENT VARIABLES IN ORDER TO FORM CROSS-PLOTS.
C
C
C
C
C

C
C
C---------[illegible] TITLI, TITLD,
C         *                                         (PTITL(I),I=1,6)
C
C                          ALPHANUMERIC TITLING INFORMATION
C                          TO INCLUDE ON PLOTTED OUTPUT.
C
C                          TITLI = INDEPENDENT VARIABLE TITLE.
C
C                          TITLD = DEPENDENT VARIABLE TITLE.
C
C                          PTITL = OVERALL TITLE FOR PARTICULAR
C                                  IDENTIFICATION OF THIS FRAME.
C
C
C         GO TO [illegible]
C
C    1000 CONTINUE
C
C         RETURN
C         END
C
C
C
C
C
C*****************************************************************************
C*
C*     [illegible]
C*
C*****************************************************************************
C
C
C      [illegible]
C
```

```
C                    LNAM = 4HTIME FOR LINEARIZED                          015738
C                              TIME RESPONSE                               015739
C                                                                          015740
C                         = 4HFREQ FOR LINEARIZED                          015741
C                              FREQUENCY DOMAIN STUDY                      015742
C                                                                          015743
C        IF (LNAM .EQ. 4HTIME) RETURN                                      015744
C                                                                          015745
C        IF (LNAM .EQ. 4H    ) RETURN                                      015746
C                                                                          015747
C                                                                          015748
C---------CALL READIM (LCRY, 9, NCYC, 9, KR)                               015749
C                                                                          015750
C              MATRIX SIZE  9 BY NCYC DEFINING FREQUENCY                   015751
C              ANALYSIS CONTROL DATA                                       015752
C                                                                          015753
C                  NCYC = NO. OF SEPARATE TRANSFER                         015754
C                         FUNCTION CYCLES TO CONSIDER                      015755
C                         FOR THIS SIMULATION                              015756
C                                                                          015757
C              FOR THE JTH CYCLE -                                         015758
C                                                                          015759
C              LCRY(1,J) - ITYPE --- INTEGER CLASSIFYING THE               015760
C                                    TRANSFER FUNCTION TYPE.               015761
C                                                                          015762
C                        ITYPE = 1 PLANT ONLY (G)                          015763
C                              = 2 CONTROLLER (H)                          015764
C                              = 3 OPEN LOOP (GH)                          015765
C                              = 4 OPEN LOOP (HG)                          015766
C                              = 5 CLOSED LOOP (G/(1+GH))                  015767
C                              = 6 CLOSED LOOP                             015768
C                              = 7 PSEUDO OPEN LOOP.                       015769
C                                 = 8 SPECIAL CASE (SEE BELOW)             015770
C                               PERMITS OPENING OF A                       015771
C                               SINGLE RETURN LOOP.                        015772
C                                                                          015773
C                               A MINUS SIGN ON ITYPE                      015774
C                               EQUAL (5,6,OR 7) INDICATES                 015775
C                               NEGATIVE CONTROLLER FEEDBACK.              015776
C                               A MINUS SIGN ON ITYPE EQUAL 8              015777
C                               INDICATES COMPLETELY CLOSED                015778
C                               LOOP TRANSFER FUNCTION.                    015779
C                                                                          015780
C                                                                          015781
C                                                                          015782
C                                                                          015783
C                                                                          015784
C              LCRY(2,J) - ITFIN --- TRANSFER FUNCTION INPUT               015785
C                                    VARIABLE IDENTIFICATION.              015786
C                                    THIS INTEGER IS A LOCAL               015787
C                                    IDENTIFICATION INTEGER.               015788
C                                    REFERENCES (DEPENDING                 015789
C                                    UPON ITYPE) EITHER A                  015790
C                                    SENSOR SIGNAL OR A                    015791
C                                    CONTROLLER OUTPUT VARIABLE            015792
C                                    WHICH IS THE VALUE OF                 015793
C                                    THE EXPRESSION --                     015794
C                                                                          015795
C                                        Y(IJT)                            015796
C                                      ----------  =  IF                   015797
```

```
                    [illegible]                                    015798
                                                                   015799
                                                                   015800
      [illegible]                                                  015801
                              [illegible] N.                       015802
                              [illegible]                          015803
                              [illegible]                          015804
                              [illegible]                          015805
                              [illegible] A                        015806
                              [illegible]                          015807
                              [illegible]                          015808
                              [illegible]                          015809
                              [illegible]                          015810
                                                                   015811
                    [illegible]                                    015812
                    [illegible]                                    015813
                    [illegible]                                    015814
                                                                   015815
      [illegible]                                                  015816
        [illegible]                                                015817
        [illegible]                                                015818
        [illegible]                                                015819
                                                                   015820
      [illegible]                                                  015821
                                                                   015822
               [illegible]                                         015823
               [illegible]                                         015824
                                                                   015825
      [illegible]                                                  015826
                                                                   015827
               [illegible]                                         015828
               [illegible]                                         015829
               [illegible] A                                       015830
               [illegible]                                         015831
               [illegible] MATRIX.                                 015832
               [illegible]                                         015833
                                                                   015834
                                                                   015835
                                                                   015836
                                                                   015837
                                                                   015838
                                                                   015839
               [illegible] PROGRAM WILL [illegible] TO             015840
                           [illegible] TRANSFORM.                  015841
                                                                   015842
                        DEFAULT VALUE IS [illegible]               015843
                      [illegible] TRANSFORM WILL BE USED.          015844
                                                                   015845
          NOTE - PROGRAM EXTRACTS ROOTS FOR BOTH                   015846
                 [illegible] TRANSFORMS. THIS SERVES               015847
                 AS A [illegible] CHECK ON THE                     015848
                 [illegible] QUALITY. ALTHOUGH IT IS A             015849
                 [illegible] ROOTS FROM AN                         015850
                 [illegible] CAN BE [illegible] THAN               015851
                 THOSE OBTAINED FROM AR.                           015852
                                                                   015853
      CARD [illegible] NO. OF [illegible] VARIABLES TO FEED BACK. (ITYPE = 7    015854
                 [illegible] VARIABLES CAN BE FED BACK FOR         015855
                 THE TYPE 7 [illegible] OPEN LOOP                  015856
                 TRANSFER FUNCTION.                                015857
```

```
C                                                                       015858
C                 IRY( 7,J) = LOCAL NO. OF FIRST  B TO RETAIN.          015859
C                                                                       015860
C                 IRY( 8,J) = LOCAL NO. OF SECOND B TO RETAIN.          015861
C                                                                       015862
C                 IRY( 9,J) = LOCAL NO. OF THIRD  B TO RETAIN.          015863
C                                                                       015864
C                                                                       015865
C----------CALL READIM (IRY, 9, NCYC, 3, KR)                            015866
C                                                                       015867
C                 MATRIX SIZE 9 BY NCYC DEFINING EXPONENT FOR           015868
C                 TOLERANCES    TOL = (10.)**EXP                        015869
C                                                                       015870
C                 FOR THE JTH CYCLE -                                   015871
C                                                                       015872
C                 IRY(1,J) - ROOT TOLERANCE EXPONENT                    015873
C                 IRY(2,J) = GAIN TOLERANCE EXPONENT                    015874
C                 IRY(3,J) - ROOT TOLERANCE EXPONENT USED TO            015875
C                                REMOVE SHIFT FREQUENCY (SUBROUTINE NUMS) 015876
C                                                                       015877
C                 NOTE -- IF ROOT OR GAIN LE TOL, THEN                  015878
C                          SET ROOT OR GAIN EQ 0.0                      015879
C                                                                       015880
C                                                                       015881
C                                                                       015882
C                                                                       015883
C                                                                       015884
C                                                                       015885
C                                                                       015886
C          DO 500 ICYC=1,NCYC                                           015887
C                                                                       015888
C          IF (ITYPE .EQ. 0) GO TO 500                                  015889
C                                                                       015890
C                                                                       015891
C----------READ(IT,FORMAT = 20A4) (TITLE(I),I=1,20)                     015892
C                                                                       015893
C                 80 CHARACTER TITLE FOR TRANSFER FUNCTION              015894
C                 IDENTIFICATION                                        015895
C                                                                       015896
C                                                                       015897
C----------READ(IT,FORMAT = 20A4) (LPNAME(I),I=1,5), LPTAPE,LEIGV,LPOLY 015898
C                                                                       015899
C                  LPNAME(I) PERMITS UP TO 5 FOUR CHARACTER             015900
C                            IDENTIFICATIONS WHICH SELECT THE           015901
C                            PLOT DISPLAY MODE.                         015902
C                                                                       015903
C                  LPNAME(I) = 4H       (ALL BLANKS) NO DISPLAYS        015904
C                                        ARE IMPLEMENTED --- GO TO 500  015905
C                                        THE CHARACTERISTIC ROOTS       015906
C                                        FOR THE SYSTEM ARE FOUND.      015907
C                                                                       015908
C                  LPNAME(I) = 4HBODE  ONLY A BODE DISPLAY.             015909
C                                                                       015910
C                            = 4HNICH  ONLY A NICHOLS DISPLAY.          015911
C                                                                       015912
C                            = 4HNYQU  ONLY A NYQUIST DISPLAY.          015913
C                                                                       015914
C                            = 4HNINY  BOTH NICHOLS AND NYQUIST.        015915
C                                                                       015916
C                            = 4HBENN  GIVES BODE, NICHOLS, NYQUIST.    015917
```

```
C
C                       = 4HRLOT  GIVES A ROOT LOCUS DISPLAY.
C
C               LPTAPE  = 4H7TRK  DUMP /PSTUFF/ ON 7-TRACK TAPE
C                                 UNIT 12 IF SO. DON'T IF.NE.4H7TRK
C
C               LEIGV   = 4HEIGV  COMPUTE EIGENVECTORS
C               LPOLY   = 4HPOLY  COMPUTE AND PRINT TRANSFER
C                                 FUNCTION AS RATIO OF POLYNOMIALS
C
C         IF(LEIGV .NE. 4HEIGV) GO TO 23
C
C----------READ(INIT,FORMAT = 2F10.0) FQMIN, FQMAX
C
C      23 CONTINUE
C
C         COMPUTE EIGENVECTORS ASSOCIATED WITH EIGENVALUES HAVING
C         IMAGINARY PART .GT. FQMIN AND .LT. FQMAX
C
C               EIGENVECTORS ASSOCIATED WITH REPEATED EIGENVALUES
C               CANNOT BE FOUND WITH PRESENT SUBROUTINE EIGVEC
C
C
C
C
C
C         [illegible]
C
C
C          IF (LPNAME(ICP) .EQ. 4H    ) GO TO 333
C
C          IF (LPNAME(ICP) .EQ. 4HCODE
C        *.OR. LPNAME(ICP) .EQ. 4HBICH
C        *.OR. LPNAME(ICP) .EQ. 4HNYQU
C        *.OR. LPNAME(ICP) .EQ. 4HSINV
C        *.OR. LPNAME(ICP) .EQ. 4HICHN) GO TO 200
C
C          IF (LPNAME(ICP) .EQ. 4HRLOT) GO TO 300
C
C
C     200 CONTINUE
C
C
C              *** FREQUENCY RESPONSE SECTION ***
C
C
C-----------READ(INIT,FORMAT = 6E12.0) FMIN, FMAX, DBMIN, DBMAX, AMIN, AMAX
C
C                 FMIN  = FREQUENCY SWEEP LOWER LIMIT
C                 FMAX  = FREQUENCY SWEEP UPPER LIMIT
C                 DBMIN = MINIMUM DB AMPLITUDE FOR BODE, NICHOLS PLOTS
C                 DBMAX = MAXIMUM DB AMPLITUDE FOR BODE, NICHOLS PLOTS
C                 AMIN  = MINIMUM AMPLITUDE FOR NYQUIST PLOTS
C                 AMAX  = MAXIMUM AMPLITUDE FOR NYQUIST PLOTS
C
C
C         GO TO 333
C
C
C     300 CONTINUE
```

```
C
C
C
C
C
C
C
C
C                 *** ROOT LOCUS SECTION ***
C
C
C----------CALL READIM (IJM, 2, NREC, 2, KR)
C
C                 MATRIX SIZE 2 BY NREC FOR ROOT LOCUS PLOT CONTROL
C
C                 NREC = NUMBER OF ROOT LOCI TO PERFORM
C
C                 FOR THE JTH ROOT LOCI -
C
C                   IJM(1,J) = ISNIM    = 1  STARTING POINT IS
C                                             OPEN LOOP ZERO.
C                                       = 2  STARTING POINT IS
C                                             OPEN LOOP POLE.
C                                       = 3  STARTING POINT IS
C                                             CLOSED LOOP POLE.
C
C
C                   IJM(2,J) = ELEMENT LOCATION IN ROOT ARRAY
C                                FOR STARTING ROOT LOCI.
C
C
C----------CALL READ (A1, 6, NREC, KR, KR)
C
C                 MATRIX SIZE 6 BY NREC FOR ROOT LOCUS CONTROL DATA
C
C                 FOR THE JTH ROOT LOCI -
C
C                   A1(1,J) = THETA(J)  INITIAL SEARCH ANGLE.
C                                NORMALLY =10). (DEGREES)
C
C                   A1(2,J) = SCL  SCALE FACTOR, NORMALLY
C                                SCL = 1.)
C
C                   A1(3,J) = ALCC  PHASE CONTROL PARAMETER.
C                                ALCC = +1. --- 180. DEG. PHASE.
C                                ALCC = -1. --- 0    DEG. PHASE.
C
C                   A1(4,J) = XMIN  MIN REAL VALUE TO PLOT.
C
C                   A1(5,J) = XMAX  MAX REAL VALUE TO PLOT.
C
C                   A1(6,J) = YMAX  MAX IMAG VALUE TO PLOT AND
C                                YMIN SET TO - YMAX.
C
C
C     NOT EXISTING
C
C        RETURN
C        RETURN
C        RETURN
```

```
C          [illegible]                                                                                  016038
C          [illegible]                                                                                  016039
C          [illegible]                                                                                  016040
C          [illegible]                                                                                  016041
        RETURN                                                                                          016042
        END                                                                                             016043






      SUBROUTINE [illegible] (*)                                                                        016044
C                                                                                                       016045
C                                                              [illegible] # 115                        016046
C                                                                                                       016047
C                                                                                                       016048
C                                                                                                       016049
C***********************************************************************                                016050
C***********************************************************************                                016051
C*                                                                                                      016052
C*       PROPERTIES OF PROGRAM SIZES & INPUT DATA                                                       016053
C*                                                                                                      016054
C***********************************************************************                                016055
C***********************************************************************                                016056
C                                                                                                       016057
C                                                                                                       016058
C  -------  [illegible] PROGRAM SIZES                                                                   016059
C                                                                                                       016060
C                        [illegible] INTO THE                                                           016061
C                        PROGRAM WITH [illegible]                                                       016062
C                        [illegible].                                                                   016063
C                                                                                                       016064
C                                                                                                       016065
C                                                                                                       016066
C***********************************************************************                                016067
C*                                                                                                      016068
C*      [illegible] OUTPUT                                                                              016069
C*                                                                                                      016070
C***********************************************************************                                016071
C                                                                                                       016072
C                                                                                                       016073
C                                                                                                       016074
C                                                                                                       016075
C----------  SUMMARY OF [illegible]-SIMULATION INPUT DATA * * * * * * * * * *                            016076
C                                                                                                       016077
C                                                                                                       016078
C----------  ACTUAL      MAXIMUM     INTEGRATION    GRAVITY GRADIENT                                    016079
C----------  SIZES       SIZES       DATA           DATA                                                016080
C----------  ------      -------     -----------    ---------------                                     016081
C                                                                                                       016082
C----------  N1     =    N1MAX  =    START1 =       G1    =  GA41   =                                   016083
C                                                                                                       016084
C----------  NM     =    NMMAX  =    DELTAT =       G2    =  GA42   =                                   016085
C                                                                                                       016086
C----------  NSPI   =    NSPMAX =    ENDTAT =       G3    =  GA43   =                                   016087
C                                                                                                       016088
C----------  NSENG  =    NSMMAX =                   GMAS  =  HGMAS  =                                   016089
C                                                                                                       016090
C----------  NDELTA =    NMAXGD =                                                                       016091
```

```
C
C---------- NB     = NBBOD  -             MISC. DATA
C                                         ----------
C---------- NDELTA = KAD    -
C
C---------- NLAM   = KY     -             NOPRNT =
C
C---------- NQ     = KQ     -             NOPLOT =
C
C----------                               IFBACK =
C
C
C
C
C
C
C
C                 THE FOLLOWING LIST IDENTIFIES THE
C                 OUTPUT VARIABLES SUMMARIZED ON THE
C                 PREVIOUS PAGE.
C
C
C*****************************************************************
C
C
C                 NB     = NO. OF BODIES.
C
C                 NH     = NO. OF HINGES.
C
C                 NSPT   = NO. OF SENSOR POINTS.
C
C                 NDELTA = NO. OF CONTROL SYSTEM DELTAS.
C
C                 NQ     = NO. OF Q'S.
C
C                 NETA   = NO. OF ETA'S.
C
C                 NLAM   = NO. OF LAMBDA'S (CONSTRAINTS).
C
C                 NEQ    = NO. OF STATE EQUATIONS.
C
C                 NBMAX  = MAXIMUM DIMENSIONED NO. OF BODIES.
C
C                 NHMAX  = MAXIMUM DIMENSIONED NO. OF HINGES.
C
C                 NSPMAX = MAXIMUM DIMENSIONED NO. OF SENSOR POINTS.
C
C                 NMWMAX = MAXIMUM DIMENSIONED NO. OF MOM. WHEELS.
C
C                 NMWBOD = MAXIMUM DIMENSIONED NO. OF MOM. WHEELS
C                                                     PER BODY.
C
C                 NMDBOD = MAXIMUM DIMENSIONED NO. OF MODES PER BODY.
C
C                 KMQ    = MAXIMUM DIMENSIONED NO. OF Q'S PER BODY,
C                        = 6 + NMDBOD + NMWBOD.
C
C                 KY     = MAXIMUM DIMENSIONED SIZE FOR STATE VECTOR.
C
C                 KQ     = MAXIMUM DIMENSIONED NO. OF Q'S.
```

```
C                   = NQMAX*(Q + NMODQD) + NM*MAX.                       016152
C                                                                        016153
C                                                                        016154
C                                                                        016155
C                                                                        016156
C                                                                        016157
C                                                                        016158
C               OUTPUT VARIABLE IDENTIFICATION SUMMARY                   016159
C                             (CONT'D)                                   016160
C                                                                        016161
C                                                                        016162
C*********************************************************************** 016163
C                                                                        016164
C                                                                        016165
C                                                                        016166
C               STARTT - START TIME FOR TIME RESPONSE.                   016167
C                                                                        016168
C               DELTAT = INTEGRATION STEP SIZE.                          016169
C                                                                        016170
C               ENDT   = END TIME FOR TIME RESPONSE.                     016171
C                                                                        016172
C                                                                        016173
C                                                                        016174
C                                                                        016175
C                                                                        016176
C                                                                        016177
C                                                                        016178
C                                                                        016179
C               G1     = X COMPONENT OF GRAVITY VECTOR. (INPUT)          016180
C                                                                        016181
C               G2     = Y COMPONENT OF GRAVITY VECTOR. (INPUT)          016182
C                                                                        016183
C               G3     = Z COMPONENT OF GRAVITY VECTOR. (INPUT)          016184
C                                                                        016185
C               GRAV   = SQRT(G1**2 + G2**2 + G3**2) --                  016186
C                        ACCELERATION OF GRAVITY                         016187
C                                                                        016188
C               GAMX   = DIRECTION COSINE (GRAVITY VECTOR AND X).        016189
C                                                                        016190
C               GAMY   = DIRECTION COSINE (GRAVITY VECTOR AND Y).        016191
C                                                                        016192
C               GAMZ   = DIRECTION COSINE (GRAVITY VECTOR AND Z).        016193
C                                                                        016194
C               RGRAV  = REFERENCE RADIUS FOR ACTIVE GRAVITY VECTOR.     016195
C                        (INPUT)                                         016196
C                                                                        016197
C               NQPRNT = MULTIPLE OF DELTAT TO PRINT TIME RESPONSE.      016198
C                                                                        016199
C               NQPLOT = MULTIPLE OF DELTAT TO WRITE PLOT TAPE.          016200
C                                                                        016201
C               IFLRSP = 0 (TRANSIENT TIME RESPONSE)                     016202
C                      = 1 (LINEAR ANALYSIS OR FREQ RESPONSE)            016203
C                                                                        016204
C                                                                        016205
                                                                         016206
                                                                         016207
C                                                                        016208
C                                                                        016209
C               THE FOLLOWING [illegible] ARE CERTAIN                    016210
C               [illegible] MODEL EQUATIONS AND ARE PRINTED              016211
```

```
C                   OUT AT THE BEGINNING OF EACH SIMULATION              016212
C                                                                        016213
C*********************************************************************** 016214
C                                                                        016215
C                                                                        016216
C---------THE TOPOLOGY ARRAY (ITOPOL) FOR THIS CASE FOLLOWS              016217
C                                                                        016218
C                    (1)  (2)  ...     (NH)                              016219
C                                                                        016220
C---------    1   1                                                      016221
C---------    2   1                                                      016222
C                                                                        016223
C                          THIS IS THE INPUT INTEGER ARRAY ITOPOL        016224
C                          (SEE PREVIOUS CHAPTER -- INPUT DATA)          016225
C                                                                        016226
C                                                                        016227
C---------THE CONSTRAINT SPECIFICATIONS FOR THIS CASE FOLLOW             016228
C                                                                        016229
C                    (1)   (2)  ...    (NH)                              016230
C                                                                        016231
C---------    1   1                                                      016232
C---------    2   1                                                      016233
C---------    3   1        THIS IS THE INPUT INTEGER ARRAY IHDATA        016234
C---------    4   1        (SEE PREVIOUS CHAPTER -- INPUT DATA)          016235
C---------    5   1                                                      016236
C---------    6   1                                                      016237
C---------    7   1                                                      016238
C                                                                        016239
C                                                                        016240
C---------THE SPECIFIED INITIAL HINGE ANGLES                             016241
C         AND DISPLACEMENTS (BETAH) FOLLOW                               016242
C                                                                        016243
C                     (1)   (2)   ...     (NH)                           016244
C                                                                        016245
C---------    1    1                                                     016246
C---------    2    1     ROWS 1-3 = HINGE ANGLES (CONSISTENT WITH        016247
C                                   (ITYPE).                             016248
C---------    .    .                                                     016249
C---------    .    .     ROWS 4-6 = HINGE DISPLACEMENTS- W RELATIVE TO P 016250
C---------    .    .                                                     016251
C---------    6    1                                                     016252
C                                                                        016253
C                                                                        016254
C                                                                        016255
C                                                                        016256
C                                                                        016257
C                                                                        016258
C                                                                        016259
C---------THE SPECIFIED INITIAL HINGE RATES (BETAHD) FOLLOW              016260
C                                                                        016261
C                     (1)   (2)   ...     (NH)                           016262
C                                                                        016263
C---------    1    1                                                     016264
C---------    2    1     ROWS 1-3 = ANGULAR RATES.                       016265
C---------    .    .                                                     016266
C---------    .    .     ROWS 4-6 = DISPLACEMENT RATES- W RELATIVE TO P. 016267
C---------    .    .                                                     016268
C---------    6    1                                                     016269
C                                                                        016270
C                                                                        016271
```

```
C----------THE NO. OF ELASTIC MODES/BODY ARRAY ([illegible]) FOLLOWS             016272
C                                                                                016273
C                      (1)    (2) ...   (NB)                                     016274
C                                                                                016275
C----------   I    I  THE JTH ENTRY IS THE NO. OF ELASTIC                         016276
C                      MODES FOR BODY J.                                         016277
C                                                                                016278
C                                                                                016279
C----------THE NO. OF PZS/RIGID [illegible] ARRAY ([illegible]) FOLLOWS           016280
C                                                                                016281
C                      (1)    (2) ...   (NB)                                     016282
C                                                                                016283
C----------   I    I          ELEMENTS ARE THE NO. OF PZS                         016284
C                             RANGE POINTS ON EACH BODY.                         016285
C                                                                                016286
C                                                                                016287
C----------THE NO. OF SENSOR POINTS/BODY ARRAY ([illegible]) FOLLOWS             016288
C                                                                                016289
C                      (1)    (2) ...   (NB)                                     016290
C                                                                                016291
C----------   I    I          ELEMENTS ARE THE NO. OF SENSOR                      016292
C                             POINTS ON EACH BODY.                               016293
C                                                                                016294
C                                                                                016295
C----------THE MOM. WHEEL/BODY TABLE ([illegible]) FOLLOWS                       016296
C                                                                                016297
C                       (1)   (2)    ...   (NB)                                  016298
C                                                                                016299
C----------                COL J = JTH BODY                                      016300
C----------   I    I ROW 1 = NO. OF MOM. WHEELS ON BODY J.                       016301
C----------   I    I ROW 2 = NO. OF VARIABLE SPEED WHEELS ON BODY J.             016302
C----------   .    . ROW 3 = SUCCESSIVE ROWS ARE THE MOMENTUM WHEEL              016303
C                            NUMBERS                                             016304
C----------   .    .  .      ON BODY J (IN ASCENDING ORDER).                     016305
C----------[illegible)  I  .                                                     016306
C                                                                                016307
C                                                                                016308
C                                                                                016309
C                                                                                016310
C                                                                                016311
C                                                                                016312
C                                                                                016313
C                                                                                016314
C----------THE STATE VECTOR LENGTH ARRAY ([illegible]) FOLLOWS                   016315
C                                                                                016316
C                       (1)   (2)   ... (2*NB + 2)                               016317
C                                                                                016318
C----------   I    I                                                             016319
C                                                                                016320
C                    THE ELEMENTS ARE THE LENGTHS OF SEGMENTS                    016321
C                    OF THE STATE VECTOR.                                        016322
C                                                                                016323
C                    ORDER IS                                                    016324
C                                                                                016325
C                         U(1),U(2),...,U(NB),X(1),...,X(NB),DLTA,DELTA          016326
C                                                                                016327
C                                                                                016328
C----------THE STATE VECTOR LOCATION ARRAY ([illegible]) FOLLOWS                 016329
C                                                                                016330
C                       (1)   (2)   ... (2*NB + 2)                               016331
```

```
C                                                                       016332
C----------     1     1                                                 016333
C                                                                       016334
C                    LEADING ELEMENT LOCATION IN THE STATE VECTOR       016335
C                    FOR THE SEGMENTS DESCRIBED IN ARRAY IELNCT.        016336
C                                                                       016337
C                                                                       016338
C----------THE SPECIFIED SENSOR POINT/BODY CORRELATION ARRAY (IPTSMB)   016339
C          FOLLOWS                                                      016340
C                                                                       016341
C                      (1)  (2)  ... (NSPT)                             016342
C                                                                       016343
C----------     1     1   THE JTH ELEMENT IS THE BODY NO. ON             016344
C                         WHICH SENSOR POINT J IS LOCATED.              016345
C                                                                       016346
C                                                                       016347
C                                                                       016348
C                                                                       016349
C                                                                       016350
C                                                                       016351
C                                                                       016352
C----------THE FOLLOWING DATA IS SPECIFIED MOM. WHEEL INFORMATION       016353
C           (IF ANY)                                                    016354
C                              AND CONTROLLER INFORMATION               016355
C                                                                       016356
C                                                                       016357
C----------THE SPECIFIED MOM. WHEEL CONTROL ARRAY (IMW) FOLLOWS         016358
C                                                                       016359
C                      (1)    (2)    ...  (NCFMO)                       016360
C                                                                       016361
C                        COL J = JTH MOMENTUM WHEEL                     016362
C----------     1     1  ROW 1 = WHEEL SENSOR POINT NO.                 016363
C----------     2     1  ROW 2 = SPIN AXIS                              016364
C----------     3     1  ROW 3 = 1  ACTIVE                              016365
C                              = 0  CONSTANT SPEED                      016366
C                                                                       016367
C                                                                       016368
C----------THE SPECIFIED MOM. WHEEL RATES AND INERTIAS (AMW) FOLLOW     016369
C                                                                       016370
C                      (1)    (2)    ...  (NCFMO)                       016371
C                                                                       016372
C                        COL J = JTH MOMENTUM WHEEL                     016373
C----------     1     1  ROW 1 = INITIAL WHEEL SPIN RATE                016374
C----------     2     1  ROW 2 = SPIN INERTIA                           016375
C                                                                       016376
C                                                                       016377
C----------THE SPECIFIED CONTROLLER INITIAL CONDITIONS                  016378
C           AND CHARACTERISTICS FOLLOW                                  016379
C             (THE FIRST NDELTA ARE INITIAL CONTROLLER STATE            016380
C              VARIABLES. THERE ARE   K ADDITIONAL PARAMETERS)          016381
C                                                                       016382
C                    THIS IS THE USER INPUT ARRAY CNTDTA.               016383
C                                                                       016384
C                    THE ADDITIONAL K PARAMETERS, IF ANY, ARE           016385
C                    AVAILABLE TO THE USER FOR USER-[illegible] DATA.   016386
C                                                                       016387
C                    THE FIRST NDELTA ENTRIES IN THIS ARRAY             016388
C                    ARE THE INITIAL CONDITIONS FOR THE                 016389
C                    CONTROL VARIABLES.                                 016390
C                                                                       016391
```

```
C
C
C*************************************************************************
C*************************************************************************
C
C
C
C
C
C*************************************************************************
C*
C*     SUBROUTINE RRIGID -- OUTPUT FOR RIGID BODY
C*
C*************************************************************************
C
C              THE FOLLOWING IS TYPICAL FOR THE ITH BODY
C
C----------     SUMMARY OF INPUT DATA FOR BODY  I  WHICH IS RIGID.
C
C----------THE 6X6 INERTIA MATRIX IS --
C
C                 (1)    (2)     (3)     (4)     (5)     (6)
C
C-------  1   I   IXX   -IXY    -IXZ      0     -MZ      MY
C-------  2   I  -IXY    IYY    -IYZ     MZ      0      -MX
C-------  3   I  -IZX   -IZY     IZZ    -MY      MX      0
C-------  4   I    0     MZ     -MY      M       0       0
C-------  5   I  -MZ     0       MX      0       M       0
C-------  6   I   MY    -MX      0       0       0       M
C
C
C---------FOR BODY  I THE REF HINGE NO. AND THE EULER ROTATION TYPE
C          APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS FOLLOWED BY
C          AN ARRAY CONTAINING EULER ANGLES (DEGREES) AND POSITION
C          VECTOR COMPONENTS (METERS) THAT POSITION THE HINGE TRIAD
C          WITH THE BODY TRIAD.
C
C                      -----  [illegible]  -----
C
C
C          IF BODY I HAS ANY OUTBOARD HINGES
C          THE FOLLOWING WILL BE PRINTED.
C
C
C---------FOR BODY  I THE OUTBOARD HINGE NO. AND THE EULER ROTATION TYPE
C          APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS FOLLOWED
C          BY AN ARRAY CONTAINING EULER ANGLES (DEGREES) AND POSITION
C          VECTOR COMPONENTS (METERS) THAT POSITION THE HINGE TRIAD
C          WITH THE BODY TRIAD.
C
C
C                      -----  [illegible]  -----
C
C
C*************************************************************************
C*************************************************************************
```

```
C                                                                        016452
C********************************************************************** 016453
C*                                                                       016454
C*     SUBROUTINE M5MODE - OUTPUT FOR FLEXIBLE BODY,                     016455
C                          CONSISTENT MASS MATRIX                        016456
C*                                                                       016457
C********************************************************************** 016458
C                                                                        016459
C                                                                        016460
C                    TYPICAL OUTPUT FOR ITH BODY                         016461
C                                                                        016462
C                                                                        016463
C          THIS SUBROUTINE PRINTS OUT SEVERAL MATRICES THAT ARE          016464
C          RELATED TO THE FORM OF THE GOVERNING EQUATIONS.               016465
C                                                                        016466
C          THE READER WILL BE REFERRED TO VOL I AND THE INPUT            016467
C          DATA STREAM FOR FURTHER CLARIFICATION.                        016468
C                                                                        016469
C----------SUMMARY OF INPUT DATA FOR BODY I WHICH IS                     016470
C          FLEXIBLE W/CONSISTENT MASS MATRIX.                            016471
C                                                                        016472
C                                                                        016473
C----------THE INPUT PARAMETERS--- IFRBM, IFDIAK, IFDIAD ARE             016474
C                                                                        016475
C                          SEE INPUT DATA FOR M5MODE                     016476
C                                                                        016477
C                                                                        016478
C----------THE DOF TABLE FOLLOWS---                                      016479
C                                                                        016480
C                          DEGREE OF FREEDOM AS INPUT                    016481
C                                                                        016482
C                          SEE INPUT DATA FOR M5MODE                     016483
C                                                                        016484
C                                                                        016485
C----------THE MODE SELECTION VECTOR FOLLOWS                             016486
C                                                                        016487
C                          MODE SELECTION VECTOR AS INPUT                016488
C                                                                        016489
C                          SEE INPUT DATA FOR M5MODE                     016490
C                                                                        016491
C                                                                        016492
C----------FOR BODY NO. I  THE POSITION VECTOR FROM THE BODY ORIGIN      016493
C                          TO JOINT K IS                                 016494
C                                                                        016495
C                          X =          Y =          Z =                 016496
C                                                                        016497
C                          WHERE    K IS JOINT COORDINATES               016498
C                          USED TO DEVELOP RIGID BODY MODES              016499
C                                                                        016500
C                          SEE INPUT  DATA FOR M5MODE                    016501
C                                                                        016502
C                                                                        016503
C                                                                        016504
C                                                                        016505
C                                                                        016506
C                                                                        016507
C----------THE CONSISTENT, REPARTITIONED MASS MATRIX IS--                016508
C                                                                        016509
C                          THIS IS THE REPARTITIONED MASS                016510
C                          MATRIX AND IS CONSISTENT WITH                 016511
```

CONTENTS

Page

PREFACE . . . . iii 1/A5

I. PROGRAM SYSTEM OVERVIEW 1 1/A7

- A. Introduction 2 1/A8
- B. Simulation Overview and Nomenclature 3 1/A9
- C. General Use Information 8 1/A14

II. PROGRAM SEGMENTATION 9 1/B1

III. DELINEATION OF USER-SUPPLIED MODULES 10 1/B2

- A. Logic Flow 11 1/B3
- B. User-Supplied Module Specifics 12 1/B4
- C. Selected Common-Block Information 17 1/B9

IV. PROGRAM INPUTS 19 1/B11

- A. Subroutines READ and READIM 19 1/B11
- B. Input Data Stream 20 1/B12

V. PROGRAM OUTPUTS 20 1/B12

VI. AUXILIARY PROGRAMS 33 1/C11

- A. REDIM—The Redimension Program 33 1/C11
- B. Support Programs for the DISCOS Flexible-Body Capability 42 1/D6

APPENDIX A—INPUT/OUTPUT SUBROUTINE EXPLANATIONS A-1 1/F12

APPENDIX B—SUPPORT PROGRAMS FOR THE DISCOS FLEXIBLE-BODY CAPABILITY B-1 2/A5

```
C                          THE MODE TABLE.                                016512
C                                                                         016513
C                                                                         016514
C----------THE REPARTITIONED MODAL MATRIX IS---                            016515
C                                                                         016516
C                          THIS IS THE REPARTITIONED MODAL                016517
C                          MATRIX. THE ROWS ARE CONSISTENT                016518
C                          WITH THE REPARTITIONED MASS MATRIX             016519
C                          AND THE COLS ARE CONSISTENT WITH               016520
C                          THE ELEMENTS OF THE MODE SELECTION             016521
C                          VECTOR.                                        016522
C                                                                         016523
C                                                                         016524
C----------THE -UNDEFORMED- INERTIA MATRIX (MO) IS---                     016525
C                                                                         016526
C                          THIS IS THE MO MATRIX NOTED AS                 016527
C                          EQ. II-67 OF VOL I                             016528
C                                                                         016529
C                                                                         016530
C----------                                                               016531
C           THERE THEN FOLLOWS MATRICES                                   016532
C                                                                         016533
C                      A     COEFFICIENTS                                 016534
C                      B     COEFFICIENTS                                 016535
C                     COFXY  COEFFICIENTS                                 016536
C                     COFXZ  COEFFICIENTS                                 016537
C                     COFYZ  COEFFICIENTS                                 016538
C                                                                         016539
C           WHICH ARE THE  ALPHA, B, AND C COEFFICIENTS                   016540
C           GIVEN AS EQ. II-88 OF VOL I                                   016541
C                                                                         016542
C                                                                         016543
C----------                                                               016544
C           THERE THEN FOLLOWS MATRICES                                   016545
C                                                                         016546
C                          C11, C22, C33,                                 016547
C                          C12, C13, C23                                  016548
C                                                                         016549
C           WHICH ARE IDENTIFIED AS EQ. II-89 OF VOL I                    016550
C                                                                         016551
C                                                                         016552
C                                                                         016553
C                                                                         016554
C                                                                         016555
C                                                                         016556
C                                                                         016557
C----------THE MODAL STIFFNESS IS --                                      016558
C                                                                         016559
C         -----  DATA HERE  -----                                         016560
C                                                                         016561
C                                                                         016562
C----------THE MODAL DAMPING MATRIX IS --                                 016563
C                                                                         016564
C         -----  DATA HERE  -----                                         016565
C                                                                         016566
C                                                                         016567
C----------                                                               016568
C          THERE FOLLOWS TWO ARRAYS CONTAINING THE                        016569
C                                                                         016570
C                          INITIAL MODAL DISPLACEMENTS AND                016571
```

```
C                          INITIAL MODAL VELOCITIES RESPECTIVELY.
C
C
C---------FOR BODY I THE P-Q HINGE NO., THE EULER ROTATION TYPE
C         AND THE JOINT NO. CORRESPONDING TO THE P-Q HINGE
C         APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS FOLLOWED BY AN
C         ARRAY CONTAINING EULER ANGLES THAT POSITION THE HINGE
C         TRIAD WRT THE BODY TRIAD
C
C                     ----  DATA HERE  -----
C
C
C         IF BODY I HAS ANY SENSOR POINTS,
C         THE FOLLOWING WILL BE PRINTED
C
C
C---------FOR BODY I THE SENSOR POINT NO., THE EULER ROTATION TYPE
C         AND THE JOINT NO. CORRESPONDING TO THE SENSOR POINT
C         APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS FOLLOWED
C         BY AN ARRAY CONTAINING EULER ANGLES THAT POSITION THE
C         SENSOR TRIAD WRT THE BODY TRIAD
C
C                     ----  DATA HERE  -----
C
C
C
C**********************************************************************
C**********************************************************************
C
C
C
C
C
C***********************************************************************
C*
C*    SUBROUTINE MSMODL - OUTPUT FOR FLEXIBLE BODY, LUMPED MASS MATRIX
C*
C***********************************************************************
C
C
C                    TYPICAL OUTPUT FOR THE ITH BODY
C
C
C         THIS SUBROUTINE PRINTS OUT SEVERAL MATRICES THAT ARE
C         RELATED TO THE FORM OF THE GOVERNING EQUATIONS.
C
C
C         THE READER WILL BE REFERRED TO VOL I AND THE INPUT
C         DATA STREAM FOR FURTHER CLARIFICATION.
C
C
C---------OUTPUT MATRICES  INERO, STATO, MASSO, DOCOEF, AOCOEF, ECOEF
C
C                          ARE THE J, -S, M, D, A, E, PARTITIONS
C                          RESPECTIVELY OF THE MATRIX MO OF EQ. II-87
C                          OF VOL I
C
C
C---------OUTPUT MATRIX    MOO IS MATRIX MO OF EQ. II-87 OF VOL I
C
```

```
C                                                                       016632
C---------OUTPUT MATRICES ACOF, BCOF, CXY, CXZ, CYZ,                    016633
C                                                                       016634
C                         ARE THE  ALPHA, B, AND C                      016635
C                         COEFFICIENTS IN THE MATRIX                    016636
C                         GIVEN AS EQ. II-88 OF VOL I                   016637
C                                                                       016638
C                                                                       016639
C                                                                       016640
C                                                                       016641
C                                                                       016642
C                                                                       016643
C                                                                       016644
C                                                                       016645
C                                                                       016646
C---------OUTPUT MATRICES C11, C22, C33,                                016647
C                         C12, C13, C23                                 016648
C                                                                       016649
C                         ARE THE CONSTITUENTS OF THE                   016650
C                         MATRIX GIVEN AS EQ. II-89 OF VOL I            016651
C                                                                       016652
C                                                                       016653
C---------OUTPUT MATRIX XEO                                             016654
C                                                                       016655
C                         CONTAINS THE INITIAL MODAL                    016656
C                         DEFLECTIONS. (AS INPUT)                       016657
C                                                                       016658
C                                                                       016659
C---------OUTPUT MATRIX XEOD                                            016660
C                                                                       016661
C                         CONTAINS THE INITIAL MODAL                    016662
C                         VELOCITIES.  (AS INPUT)                       016663
C                                                                       016664
C                                                                       016665
C---------FOR BODY I THE P-Q HINGE NO., THE EULER ROTATION TYPE         016666
C         AND THE JOINT NO. CORRESPONDING TO THE P-Q HINGE              016667
C         APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS FOLLOWED BY AN 016668
C         ARRAY CONTAINING EULER ANGLES THAT POSITION THE HINGE         016669
C         TRIAD WRT THE BODY TRIAD                                      016670
C                                                                       016671
C                    ----  DATA HERE  -----                             016672
C                                                                       016673
C                                                                       016674
C         IF BODY I HAS ANY SENSOR POINTS,                              016675
C         THE FOLLOWING WILL BE PRINTED                                 016676
C                                                                       016677
C                                                                       016678
C---------FOR BODY I THE SENSOR POINT NO., THE EULER ROTATION TYPE      016679
C         AND THE JOINT NO. CORRESPONDING TO THE SENSOR POINT           016680
C         APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS FOLLOWED       016681
C         BY AN ARRAY CONTAINING EULER ANGLES THAT POSITION THE         016682
C         SENSOR TRIAD WRT THE BODY TRIAD                               016683
C                                                                       016684
C                    ----  DATA HERE  -----                             016685
C                                                                       016686
C                                                                       016687
C                                                                       016688
C*********************************************************************** 016689
C*********************************************************************** 016690
C                                                                       016691
```

```
C                                                                       016692
C                                                                       016693
C                                                                       016694
C                                                                       016695
C          IF THE USER UTILIZES POLYNOMIAL INPUT FOR                    016696
C          CONTROL SYSTEM TRANSFER FUNCTIONS, THE FOLLOWING             016697
C          MATRIX WILL BE PRINTED.                                      016698
C                                                                       016699
C                                                                       016700
C---------OUTPUT MATRIX CPLY(KPLY,2*NPLY)                               016701
C                                                                       016702
C                WHERE KPLY = ROW DIMENSION SIZE OF CPLY IN             016703
C                             SUBROUTINE CONTRL                         016704
C                                                                       016705
C                      NPLY = NO. OF INPUT POLYNOMIAL RATIOS            016706
C                                                                       016707
C                AND FOR I = ODD INTEGER --                             016708
C                                                                       016709
C                    COL I   = DENOMINATOR POLYNOMIAL COEFFICIENTS      016710
C                              IN ASCENDING ORDER                       016711
C                                                                       016712
C                    COL I+1 = NUMERATOR POLYNOMIAL COEFFICIENTS        016713
C                              IN ASCENDING ORDER                       016714
C                                                                       016715
C                                                                       016716
C---------THE FOLLOWING INTEGER ARRAY (INDEP) PRESCRIBES INDEPENDENT    016717
C          VARIABLES (1) AND DEPENDENT VARIABLES (0)                    016718
C                                                                       016719
C                            THE ELEMENTS OF THIS ARRAY IDENTIFY        016720
C                            WHICH VARIABLES SURVIVE IN THE FINDD       016721
C                            SEARCH TO DETERMINE AN INDEPENDENT         016722
C                            SET TO BE INTEGRATED.                      016723
C                                                                       016724
C                            THE SURVIVING VARIABLES WILL ALSO          016725
C                            REPRESENT THE FIRST NX ROWS OF             016726
C                            THE LINEARIZED MATRIX, A, USED             016727
C                            BY SUBROUTINE DYNSUD.                      016728
C                                                                       016729
C                            NOTE -  NX = NO. OF NON-ZERO ENTRIES       016730
C                                         IN ARRAY INDEP                016731
C                                                                       016732
C                                                                       016733
C                                                                       016734
C                                                                       016735
C                                                                       016736
C*********************************************************************  016737
C*********************************************************************  016738
C                                                                       016739
C                                                                       016740
C                                                                       016741
C                                                                       016742
C                                                                       016743
C*********************************************************************** 016744
C*                                                                      016745
C*    SUBROUTINE DYNS20 OUTPUT                                          016746
C*                                                                      016747
C*********************************************************************** 016748
C                                                                       016749
C                                                                       016750
C                THE PRINTOUT IS TYPICAL FOR A GIVEN SIMULATION TIME, T.  016751
```

```
C                                                                       016752
C              THE T=0  PRINT OUT IS ALWAYS GIVEN                       016753
C                     (EVEN FOR A LINEARIZED ANALYSIS)                  016754
C                                                                       016755
C                                                                       016756
C              THE DATA ARE PRESENTED IN VECTOR FORM AND                016757
C              ORDERED AS FOLLOWS---                                    016758
C                                                                       016759
C---------THE STATE VECTOR Y =                                          016760
C                                                                       016761
C---------THE STATE VECTOR TIME DERIVATIVE YDT =                        016762
C                                                                       016763
C---------THE ETAS (EULER ANGLES, POSITION COORDINATES) ARE             016764
C                                                                       016765
C---------THE ETA TIME DERIVATIVES ARE                                  016766
C                                                                       016767
C---------THE DELTAS (CONTROL SYSTEM VARIABLES) ARE                     016768
C                                                                       016769
C---------THE DELTA TIME DERIVATIVES ARE                                016770
C                                                                       016771
C                                                                       016772
C                                                                       016773
C                                                                       016774
C                                                                       016775
C                                                                       016776
C                                                                       016777
C                                                                       016778
C---------                                                              016779
C           THE FOLLOWING INFORMATION IS TYPICAL FOR BODY I.            016780
C                                                                       016781
C                                                                       016782
C---------FOR BODY I  THE VELOCITIES ARE                                016783
C                                                                       016784
C            ORDER IS    OMGX                                           016785
C                        OMGY    BODY ANGULAR VELOCITY                  016786
C                        OMGZ                                           016787
C                                                                       016788
C                           U                                           016789
C                           V    BODY TRANSLATIONAL VELOCITY            016790
C                           W                                           016791
C                                                                       016792
C                        XID(1)                                         016793
C                        XID(2)                                         016794
C                          .     BODY MODAL VELOCITIES                  016795
C                          .                                            016796
C                          .                                            016797
C                        XID(NE)                                        016798
C                                                                       016799
C                       THETAD(I)                                       016800
C                          .     MOMENTUM WHEEL ANGULAR VELOCITY        016801
C                          .     (RELATIVE TO SENSOR POINT TRIAD)       016802
C                          .          (ACTIVE WHEEL)                    016803
C                                                                       016804
C                                                                       016805
C---------FOR BODY I THE CORRESPONDING MOMENTA ARE                      016806
C                                                                       016807
C              ORDER IS    HX                                           016808
C                          HY  BODY AXES REF. ANGULAR MOMENTUM          016809
C                          HZ   INCLS CONTRIBUTION OF CONST. SPEED WHEEL 016810
C                                                                       016811
```

```
C                            LX
C                            LY   BODY AXES REF. LINEAR MOMENTUM
C                            LZ
C
C                          P XI(1)
C                          P XI(2)
C                            .    BODY AXES REF. MODAL MOMENTUM
C                            .
C                            .
C                          P XI(NE)
C
C                          H HW(1)
C                            .    BODY AXES REF. MOM WHEEL MOMENTUM
C                            .    (ACTIVE WHEEL)
C                            .
C
C
C
C
C
C
C
C
C----------FOR BODY I ITS CONTRIBUTION TO TOTAL ANGULAR
C                      AND LINEAR MOMENTUM IS
C
C              ORDER IS     HX
C                            HY  ANGULAR MOMENTUM
C                            HZ
C                            (REFERENCED TO INERTIAL ORIGIN)
C                            LX
C                            LY  LINEAR MOMENTUM
C                            LZ
C
C
C----------FOR BODY I ITS CONTRIBUTION TO TOTAL KINETIC AND
C                      POTENTIAL ENERGIES IS
C
C
C----------FOR BODY I   THE ELASTIC DEFLECTIONS ARE
C
C
C----------FOR BODY I    TRANSFORMATION MATRIX TO BODY I COORDINATES IS
C
C                        USED TO TRANSFORM VECTORS AND TENSORS
C                        FROM BODY I FIXED REFERENCE
C                          TO BODY 1 FIXED REFERENCE
C
C
C----------FOR BODY I    CM LOCATION WRT BODY 1 REFERENCE POINT AND
C                        COORDINATES IS
C
C
C----------FOR BODY I    INERTIA TENSOR RELATIVE TO ITS CENTER OF MASS
C                        WRT BODY I COORDINATES IS
C
C
C----------THE INTERCONNECTION CONSTRAINT FORCES (LAMBDAS) ARE
C
C
```

```
C---------THE SYSTEM CENTER OF MASS RELATIVE TO BODY 1 REFERENCE POINT
C                    AND COORDINATES IS
C
C
C---------THE SYSTEM INERTIA TENSOR RELATIVE TO SYSTEM CENTER OF MASS
C                    IN BODY 1 COORDINATES IS
C
C
C---------THE TOTAL ANGULAR MOMENTUM VECTOR IS
C
C                 X,Y,Z  COMPONENTS IN INERTIAL REF. AXIS SYSTEM.
C
C
C---------THE TOTAL LINEAR MOMENTUM VECTOR IS
C
C                 X,Y,Z  COMPONENTS IN INERTIAL REF. AXIS SYSTEM.
C
C
C---------THE TOTAL ANGULAR MOMENTUM =
C---------THE TOTAL LINEAR MOMENTUM  =
C---------THE TOTAL KINETIC ENERGY   =
C---------THE TOTAL POTENTIAL ENERGY =
C---------THE TOTAL ENERGY (T+V)     =
C
C
C
C**************************************************************
C**************************************************************
C
C
C
C
C
C
C***************************************************************
C*
C*     SUBROUTINE DYNS30 OUTPUT
C*
C***************************************************************
C
C
C
C---------             SUMMARY OF PLOTTING INFORMATION
C
C
C                      THE OUTPUT SUMMARIZES THE INPUT DATA
C                      WHICH CONTROLLED THE TIME RESPONSE
C                      PLOT LOGIC.
C
C                         SEE INPUT DATA FOR DYNSCO FOR
C                         DETAILED DESCRIPTIONS.
C
C
C                  NOTE- THE OUTPUT VARIABLE NRPLOT
C                        IS THE NO. OF LOGICAL RECORDS
C                        WRITTEN ON THE PLOT TAPE.
C
C
C
C
```

```
C****************************************************************
C****************************************************************
C
C
C
C
C
C
C******************************************************************
C*
C*  SUBROUTINE DYNS40 OUTPUT - LINEARIZED SYSTEM ANALYSIS
C*
C******************************************************************
C
C
C---------OUTPUT MATRIX -A- (NJQ,NX)
C
C                         NJQ = TOTAL NO. OF EQUATIONS
C                               LINEARIZED (INCLUDING
C                               AUXILIARY EQUATIONS FROM
C                               SUBROUTINE EQADJ)
C
C                         NX  = NO OF INDEPENDENT STATE EQUATIONS
C                               DETERMINED BY FINDJ IN DYNS20.
C
C
C                         THIS IS THE MATRIX OF PARTIAL DERIVATIVES
C                         WHICH ARE THE LINEARIZED COMPONENTS OF THE
C                         STATE VARIABLES AS DETERMINED BY
C                         SUBROUTINE LINEAR.
C
C
C                         THE ORDER OF THE VARIABLES FOR THE
C                         ROWS IS
C
C                                             SIZE
C
C                             PLANT VARIABLES     NY
C
C                             CONTROL VARIABLES  NDELTA
C
C                             CONTROLLER OUTPUTS NOTQ
C
C                             SENSOR SIGNALS      NSSS
C
C
C
C                         THE COLUMN ORDER IS
C
C                             PLANT VARIABLES     NY
C
C                             CONTROL VARIABLES  NDELTA
C
C
C
C
C
C
C---------OUTPUT MATRIX -T- (NX,NX)
```

```
C
C                          THIS IS THE SIMILARITY
C                          TRANSFORMATION MATRIX
C                          THAT INTRODUCES THE (NBTQ + NASS)
C                          AUXILIARY VARIABLES INTO THE
C                          TRANSFORMED STATE EQUATIONS.
C
C                          REF. MATRIX T IN EQ. III-24 OF VOL I
C
C
C---------OUTPUT MATRIX  Y* (1,NX)
C
C                        THIS IS THE TRANSFORMED STATE
C                        VECTOR INITIAL CONDITIONS.
C
C                        REF. VECTOR Z IN EQ. III-24 OF VOL I
C
C
C---------OUTPUT MATRIX A* (NX,NX)
C
C                        THIS IS THE TRANSFORMED, LINEARIZED
C                        STATE VARIABLE COEFFICIENT MATRIX
C                        THAT IS THE BASIS FOR THE ENTIRE
C                        LINEARIZATION PACKAGE.
C
C                        REF. EQ. III-23 OF VOL I
C
C                        THE ROW/COL VARIABLE ORDERING AND
C                        SIZES ARE:
C
C                          VARIABLE ID.                 SIZE
C                          ------------                 ----
C
C                          PLANT VARIABLES               NY2
C
C                          PLANT SENSOR SIGNALS          NASS
C
C                          CONTROL SYSTEM VARIABLES      ND2
C
C                          CONTROL OUTPUTS (B'S)         NBTQ
C
C
C                   NOTE-  NY2 = NY - NXSS
C
C                          ND2 = NDELTA - NBTQ
C
C
C
C
C
C
C----------            RT A                          RTA*
C
C            REAL PART  IMAGINARY PART      REAL PART  IMAGINARY PART
C
C
C                          COMPLEX ROOTS OBTAINED FROM
C                             A AND A* RESPECTIVELY.
C
C
```

```
C                             THESE ARE THE POLES OF                    017052
C                             THE CLOSED LOOP SYSTEM.                   017053
C                                                                       017054
C                                                                       017055
C*********************************************************************  017056
C                                                                       017057
C                                                                       017058
C              THE FOLLOWING OUTPUTS ARE CHARACTERISTIC OF              017059
C              A SINGLE TRANSFER FUNCTION FREQUENCY RESPONSE.           017060
C                                                                       017061
C                                                                       017062
C---------OUTPUT MATRIX -AR-                                            017063
C                                                                       017064
C                         THIS IS THE REDUCED A* MATRIX                 017065
C                         FOR A PARTICULAR USER SPECIFIED               017066
C                         TRANSFER FUNCTION TYPE.                       017067
C                                                                       017068
C                         REF. INPUT DATA LIST AND                      017069
C                         SECTION III. D-2. IN VOLUME I.                017070
C                                                                       017071
C                         THE ROOTS OF AR  ARE THE                      017072
C                         TRANSFER FUNCTION POLES.                      017073
C                                                                       017074
C                                                                       017075
C---------OUTPUT MATRIX BCOL                                            017076
C                                                                       017077
C                         THIS IS THE VECTOR (BCOL) WITH WHICH          017078
C                         AR IS AUGMENTED TO DETERMINE THE              017079
C                         TRANSFER FUNCTION ZEROS.                      017080
C                                                                       017081
C                                                                       017082
C                                                                       017083
C                                                                       017084
C                                                                       017085
C                                                                       017086
C                                                                       017087
C---------         R AR                          RART                   017088
C                                                                       017089
C         REAL PART  IMAGINARY PART     REAL PART  IMAGINARY PART       017090
C                                                                       017091
C                                                                       017092
C                         THESE ARE THE COMPLEX ROOT ARRAYS             017093
C                         AS EXTRACTED FROM MATRICES                    017094
C                         AR AND AR TRANSPOSE RESPECTIVELY.             017095
C                                                                       017096
C                         THE USER SELECTS VIA INPUT WHICH              017097
C                         SET OF ROOTS TO USE FOR THE POLES.            017098
C                                                                       017099
C                                                                       017100
C---------         NUM                           DEN                    017101
C                                                                       017102
C         REAL PART  IMAGINARY PART     REAL PART  IMAGINARY PART       017103
C                     .                             .                   017104
C                     .                             .                   017105
C                     .                             .                   017106
C                                                                       017107
C             NUMERATOR ROOTS                  DENOMINATOR ROOTS        017108
C                                                                       017109
C          (TRANSFER FUNCTION ZEROS)      (TRANSFER FUNCTION POLES)     017110
C                                                                       017111
```

```
C                       .                               .
C                       .                               .
C                       .                               .
C
C
C
C
C
C----------OUTPUT MATRIX ROOTRE
C
C                           THE REAL PARTS OF THE COMPLEX EIGENVALUES
C                           FOR WHICH EIGENVECTORS WILL BE COMPUTED
C
C----------OUTPUT MATRIX ROOTIM
C
C                           THE IMAGINARY PARTS OF THE COMPLEX EIGENVALUES
C                           FOR WHICH EIGENVECTORS WILL BE COMPUTED
C
C                THE EIGENVALUES FOR WHICH THE ASSOCIATED
C                EIGENVECTORS ARE TO BE COMPUTED CORRESPOND
C                TO THE COMPLEX ROOTS OF THE CHARACTERISTIC
C                MATRIX (OUTPUT MATRIX -AR- )
C
C
C----------EIGENVALUE =
C
C----------EIGENVECTOR OF NORMAL MATRIX
C
C                THE SUBROUTINE USED TO COMPUTE EIGENVECTORS
C                ACCEPTS A SINGLE COMPLEX EIGENVALUE (LAMBDA),
C                THE CHARACTERISTIC MATRIX AR AND THE MATRIX AR*AR.
C                 IF LAMBDA IS A MULTIPLE ROOT THE ERROR MESSAGE
C            '****WARNING*** ZEROS ON DIAGONAL OF FACTORED MATRIX.
C                            CHECK FOR MULTIPLE EIGENVALUES'
C                 IS PRINTED; OTHERWISE
C                THE NA ELEMENTS OF THE EIGENVECTOR XR OF THE NORMAL
C                MATRIX ARE FOUND; THAT IS
C                        AR*XR = LAMBDA*XR
C
C
C
C
C
C
C----------OUTPUT MATRIX RRED
C
C
C                TRANSFER FUNCTION ROOT ARRAY CONTAINING ROOT
C                COUNTS, TIME CONSTANTS, DAMPING, AND FREQUENCY
C                FOR ZEROS AND POLES.
C
C
C                    ELE 1 = NO. OF NUMERATOR REAL ROOTS.      NNR
C
C                    ELE 2 = NO. OF NUMERATOR COMPLEX PAIRS.   NNC
C
C                    ELE 3 = NO. OF NUMERATOR PURE S'S.        NNZ
C
C                    ELE 4 = NO. OF DENOMINATOR REAL ROOTS.    NDR
C
```

```
C                   ELE 5 = NO. OF DENOMINATOR COMPLEX PAIRS,NDC
C
C                   ELE 6 = NO. OF DENOMINATOR FREE S'S.        NDZ
C
C                   ELE 7 = BODE GAIN, KB.
C
C
C                          THE ROOTS FOLLOW IN THE ORDER
C
C
C            ELEMENT LOCATION         DESCRIPTION
C            ----------------         -----------
C
C              6 THRU NNR+7           NUMERATOR TIME CONSTANTS
C
C           NNR+8 THRU 2*NNC + 7      NUMERATOR DAMPING AND FREQUENCIES
C
C                                     ZETA1, OMEGA1, ZETA2, OMEGA2 ---
C                          FREE S'S ARE NOT INCLUDED
C
C
C              REMAINING ELEMENTS ARE DENOMINATOR ROOTS IN SAME
C              ORDER AS NUMERATOR ROOTS, IE.,
C
C           TAU(1),...,TAU(NDR),ZETA(1),OMEGA(1),...,ZETA(NDC),OMEGA(NDC)
C
C
C
C
C
C
C
C              THE TRANSFER FUNCTION FREQUENCY RESPONSE FOLLOWS
C
C
C--------- FREQ/RAD/SEC  FREQ/HERTZ  REAL  IMAG  AMP  DECIBELS  RAD  DEG
C                     .                .                .
C                     .                .                .
C                     .                .                .
C
C
C              THE DAMPED RESONANCES (BASED ON A
C              POLE OR ZERO) ARE IDENTIFIED
C              BY  *****  IN BOTH THE LEFT AND RIGHT MARGINS.
C
C
C              ******************************************
C
C
C             THE FOLLOWING OUTPUTS ARE TYPICAL OF A ROOT
C             LOCUS INVESTIGATION.
C
C
C---------OUTPUT MATRICES   PDEN
C                           PNUM
C
C                          THE DENOMINATOR AND NUMERATOR
C                          TRANSFER FUNCTION POLYNOMINAL
C                          COEFFICIENTS --- ASCENDING POWERS OF S.
C---------
```

```
C             N(S) =        NUMERATOR                                                017232
C                              AND                                                   017233
C             D(S) =        DENOMINATOR                                              017234
C                                                                                    017235
C                UNPROCESSED                                                         017236
C                POLYNOMIAL COEFFICIENTS AS USED BY ROOTS.                           017237
C                                                                                    017238
C                                                                                    017239
C----------  STARTING POINT =  USER IDENTIFIED STARTING POINT                        017240
C                               FOR THE LOCI.                                        017241
C                                                                                    017242
C----------  SCAN LIMITS    =   LIMITS ON REAL AND IMAGINARY                         017243
C                               COMPONENTS FOR THE LOCI.                             017244
C                                                                                    017245
C                                                                                    017246
C----------    GAIN            ROOTS            ERROR                                017247
C                                                                                    017248
C                                                                                    017249
C                          ROOT LOCUS OUTPUT                                         017250
C                                                                                    017251
C                                                                                    017252
C                                                                                    017253
C*********************************************************************               017254
C*********************************************************************               017255
      RETURN                                                                         017256
      END                                                                            017257



      SUBROUTINE DEFS                                                                017258
C                                                             DEBUG * 116            017259
C                                                                                    017260
C                                                                                    017261
C                                                                                    017262
C                        DEFINITIONS OF SYMBOLIC NAMES                               017263
C                            USED IN DISCOS                                          017264
C                                                                                    017265
C                                                                                    017266
C                                                                                    017267
C      MAXIMUM SIMULATION MODEL SIZE PARAMETERS                                      017268
C-------------------------------------------                                         017269
C                                                                                    017270
C         THESE PARAMETERS ARE INTEGER*4 CONSTANTS FOR ANY PARTICULAR                017271
C         COMPILED VERSION OF DISCOS. THEY MAY BE ALTERED ONLY VIA USE               017272
C         OF THE REDIMENSION PROGRAM                                                 017273
C                                                                                    017274
C      THE FOLLOWING ARE DEFINED IN MAIN, STORED IN /MAXNUM/ AND PRINTED             017275
C       ALONG WITH THE ECHO OF THE INPUT DATA                                        017276
C                                                                                    017277
C      NBMAX  = MAXIMUM NUMBER OF BODIES (RIGID + FLEXIBLE)                          017278
C      NHMAX  = MAXIMUM NUMBER OF HINGE POINTS IN SIMULATION                         017279
C      NSPMAX = MAXIMUM NUMBER OF SENSOR POINTS IN SIMULATION                        017280
C      NMWMAX = MAXIMUM NUMBER OF MOMENTUM WHEELS IN SIMULATION                      017281
C      NMWBOD = MAXIMUM NUMBER OF MOMENTUM WHEELS IN ANY BODY                        017282
C      NMOBOD = MAXIMUM NUMBER OF FLEXIBLE BODY MODES ALLOWED PER BODY               017283
C      KMU    = MAXIMUM NUMBER OF VELOCITY COORDINATES PER BODY                      017284
C                     KMU = 6 + NMOBOD + NMWBOD                                      017285
```

```
C       KY      = MAXIMUM NUMBER OF PLANT + CONTROL SYSTEM STATE VARIABLES
C                 ALLOWED FOR NON-LINEAR TIME RESPONSE ANALYSIS
C       KU      = MAXIMUM NUMBER OF VELOCITY COORDINATES IN SIMULATION
C                         KU = NBMAX*(6 + NMCBOD) + NMAMAX
C
C       THE FOLLOWING ARE STORED IN OTHER COMMON BLOCKS BUT ARE NOT ECHOED
C        WITH THE INPUT DATA
C
C       KSUMRY = MAX0(NMAMAX,NSPMAX) STORED IN /SUMMRY/
C       KR      = MAXIMUM NUMBER OF INDEPENDENT LINEARIZED EQUATIONS WHICH
C                 MAY BE SUBJECTED TO FREQUENCY DOMAIN ANALYSIS, STORED
C                 IN /LDSIZE/
C       KRT     = 2*KR + 7  (STORED IN /LDSIZE/)
C       KRX     = 2*KRT      (STORED IN /LDSIZE/)
C       KV1     = 2*KRT      (STORED IN /LDSIZE/)
C       KV2     = KRT        (STORED IN /LDSIZE/)
C       KVX     = KR         (STORED IN /LDSIZE/)
C
C       THE FOLLOWING ARE NOT STORED IN DISCOS BUT ARE DEFINED IN THE
C          REDIMENSION PROGRAM AND USED THERIN TO SET DISCOS DIMENSIONS
C
C       IGASMX = MAXIMUM NUMBER OF EQUATIONS WHICH MAY BE NUMERICALLY
C                LINEARIZED
C       MAXCNT = MAXIMUM NUMBER OF CONSTANTS WHICH MAY BE INPUTTED BY
C                THE USER IN DYNS10
C                      MAXCNT = KCONT (IN DYNS10)
C       JMAXC  = MAXIMUM NUMBER OF JOINTS (GRID POINTS) ON ANY FLEXIBLE
C                BODY TO BE DESCRIBED BY CONSISTANT MASS MATRIX DATA
C                      KJDCF  = JMAXC  (IN MSMODC)
C       JMAXL  = MAXIMUM NUMBER OF JOINTS (GRID POINTS) ON ANY FLEXIBLE
C                BODY TO BE DESCRIBED BY LUMPED PARAMETER DATA
C                      KJOINT = JMAXL  (IN MSMODL)
C       MLTAMX = MAXIMUM ALLOWABLE ROW DIMENSION FOR B MATRIX IN MULTA
C                      MLTAMX (SHOULD BE) = MAX0(6*JMAXC,IGASMX)
C       MXBTAB = MAXIMUM MATRIX DIMENSION IN BTABA
C                      MXBTAB (SHOULD BE) = 6*JMAXC
C       MLT3MX = MAXIMUM MATRIX DIMENSION IN MULT3
C       MACMAX = MAXIMUM MATRIX DIMENSION IN MULTAD
C       MAXZP  = ROW DIMENSION FOR TIME HISTORY PLOT DATA ARRAY
C                      MAXZP = KRPLOT (IN DYNS30)
C       MXJVPL = SIZE OF TIME HISTORY PLOT VARIABLE SELECTION VECTOR
C                      MXJVPL = KCPLOT (IN DYNS30), DON'T CHANGE
C       MAXDUM = SIZE OF DUMMY VECTOR DUM IN DYNS30
C       L1      = USED IN REDIMENSION PROGRAM TO SET PARAMETER KR IN
C                 DYNS40, DEFINED ABOVE
C       L5      = MAXIMUM SIZE OF FREQUENCY RESPONSE DATA VECTORS
C                 IN /PSTUFF/
C                      L5 = KUSAVE (IN SFEQ2)
C
C
C
C       BASIC SIMULATION MODEL CREATED BY THE USER
C-----------------------------------------
C
C           PLANT SIZE:     (THESE PARAMETERS ARE INPUTTED VIA DYNS10,
C           ----------       ECHOED IN THE STANDARD PRINTOUT, STORED
C                            IN /SPECIF/ AND ARE INTEGER*4 CONSTANTS)
C
C       NB      = TOTAL NUMBER OF RIGID AND FLEXIBLE BODIES
C                      (NB MUST BE .GE.2)
```

```
C     NH     = TOTAL NUMBER OF HINGES. HINGE 1 ALWAYS TAKEN TO BE              017346
C              BETWEEN BODY 1 AND THE INERTIAL REFERENCE                        017347
C     NSPT   = TOTAL NUMBER OF SENSOR POINTS. ALL MOMENTUM WHEELS ARE          017348
C              LOCATED AT SENSOR POINTS. ALL POINTS AT WHICH KINEMATIC         017349
C              DATA MUST BE OBTAINED ARE SENSOR POINTS (HINGE POINTS           017350
C              NOT INCLUDED)  (NSPT MUST BE .GE.1)                             017351
C     NOFMO  = TOTAL NUMBER OF CONSTANT AND VARIABLE SPEED WHEELS              017352
C                                                                              017353
C        CONTROL SYSTEM SIZE    (THE USER MAY REQUIRE THAT SEVERAL             017354
C        -------------------     DIFFERENTIAL EQUATIONS BE INTEGRATED          017355
C                                TO DEFINE EFFECTS ACTING ON THE SYSTEM)       017356
C                                                                              017357
C     NDELTA = TOTAL NUMBER OF USER DEFINED FIRST ORDER DIFFERENTIAL           017358
C              EQUATIONS CODED IN SUBROUTINE CONTRL. AN INTEGER*4              017359
C              CONSTANT STORED IN /SPECIF/ INPUTTED VIA DYNS10                 017360
C                                                                              017361
C        PLANT DESCRIPTION :  (OBTAINED VIA INPUT DATA CALLS FROM              017362
C        -----------------      DYNS10, ECHOED IN STANDARD PRINTOUT AND        017363
C                               STORED IN EITHER /SPECIF/ OR /NHNS/            017364
C                                                                              017365
C     ITOPOL = PLANT TOPOLOGY, AN INTEGER*4 ARRAY DIMENSIONED (2,NHMAX)        017366
C              USED ALSO TO DEFINE POSITIVE DIRECTION OF RELATIVE MOTION       017367
C                         AT HINGE POINT J, J=1,2,...,NH                       017368
C                BODY NUMBER N=ITOPOL(1,J), WHICH CONTAINS THE Q-FRAME,        017369
C                         IS CONTIGUOUS TO                                     017370
C                BODY NUMBER M=ITOPOL(2,J), WHICH CONTAINS THE P-FRAME,        017371
C              BY DEFINITION:                                                  017372
C                             HINGE POINT 1 IS BODY 1 REFERENCE POINT          017373
C                             FOR J=1, N=1 AND M=0; ALSO,                      017374
C                             THE MOTION OF THE Q-FRAME WILL BE MEASURED       017375
C                             RELATIVE TO THE P-FRAME AT EVERY HINGE           017376
C                             POINT IN THE SIMULATION                          017377
C     IRGFLX = BODY TYPE, AN INTEGER*4 ARRAY DIMENSIONED (NBMAX)               017378
C                 IRGFLX(N) = 0 IMPLIES THAT BODY N IS A RIGID BODY            017379
C                 IRGFLX(N) = M IMPLIES THAT BODY N IS A FLEXIBLE BODY         017380
C                                AND THAT M MODES OF VIBRATION ARE TO BE       017381
C                                USED TO COMPUTE ITS FLEXURAL DEFORMATION      017382
C     IMO    = MOMENTUM WHEELS, INTEGER*4 ARRAY DIMENSIONED (3,NMWMAX)         017383
C                     FOR MOMENTUM WHEEL NUMBER J, J=1,2,...,NOFMO             017384
C                 IMO(1,J) = SENSOR POINT NUMBER ASSOCIATE WITH WHEEL J        017385
C                 IMO(2,J) = THE AXIS NUMBER (1,2 OR 3) OF THE SENSOR          017386
C                            POINT REFERENCE FRAME WHICH IS ALIGNED WITH       017387
C                            SPIN AXIS OF THE WHEEL J                          017388
C                 IMO(3,J) = 1 IF WHEEL J IS ACTIVE OR VARIABLE SPEED          017389
C                            0 IF WHEEL J IS CONSTANT SPEED                    017390
C     NMOW   = MOMENTUM WHEEL/BODY TABLE, COMPUTED INTEGER*4 ARRAY             017391
C              DIMENSIONED (2+NMWBOD,NBMAX). FOR BODY N, N=1,2,...,NB          017392
C               NMOW(1,N) = NUMBER OF MOMENTUM WHEELS ON BODY N                017393
C               NMOW(2,N) = NUMBER OF VARIABLE SPEED WHEELS ON BODY N          017394
C               NMOW(3,N) = LOWEST MAGNITUDE WHEEL NUMBER ON BODY N            017395
C                   .       = NEXT TO LOWEST WHEEL NUMBER ON BODY N            017396
C                   .       = ETC.                                             017397
C                   .       =                                                  017398
C               NMOW(L,N) = HIGHEST MAGNITUDE WHEEL NUMBER ON BODY N           017399
C                            WHERE L=2+NMOW(1,N)                               017400
C     IFTSMX = SENSOR POINT/BODY TABLE, INPUTTED INTEGER*4 ARRAY               017401
C              DIMENSIONED (NSPMAX)                                            017402
C               IFTSMW(J) = BODY NUMBER OF THE BODY ON WHICH                   017403
C                           SENSOR POINT J IS LOCATED J=1,2,...,NSPT           017404
C     NHBDI  = HINGE/BODY TABLE, COMPUTED INTEGER*4 ARRAY                      017405
```

```
C              DIMENSIONED (NBMAX)
C               NHPOI(N)  = NUMBER OF HINGE POINTS ON BODY N,N=1,....,NB
C     NSPOI  = SENSOR/BODY TABLE, COMPUTED INTEGER*4 ARRAY
C              DIMENSIONED (NBMAX)
C               NSPOI(N)  = NUMBER OF SENSOR POINTS ON BODY N,N=1,....,NB
C     IHDATA = HINGE POINT CONSTRAINT AND EULER ANGLE TYPE, INPUTTED
C              INTEGER*4 ARRAY DIMENSIONED (7,NHMAX)
C                   FOR HINGE POINT J, J=1,2,....,NH
C               IHDATA(1,J) = ITYPE (CODE NUMBER ASSIGNED TO THE DESIRED
C                             EULER ANGLE ROTATION SEQUENCE TO BE USED
C                             TO ORIENT G-FRAME WRT P-FRAME); THAT IS
C                             ITYPE =1-(1,2,3)    ITYPE =  7-(2,1,2)
C                             ITYPE =2-(1,2,1)    ITYPE =  8-(2,1,3)
C                             ITYPE =3-(1,3,1)    ITYPE =  9-(3,1,2)
C                             ITYPE =4-(1,3,2)    ITYPE = 10-(3,1,3)
C                             ITYPE =5-(2,3,1)    ITYPE = 11-(3,2,3)
C                             ITYPE =6-(2,3,2)    ITYPE = 12-(3,2,1)
C               IHDATA(2,J) = CODE TO DEFINE THETA 1 ROTATION CONSTRAINT
C               IHDATA(3,J) = CODE TO DEFINE THETA 2 ROTATION CONSTRAINT
C               IHDATA(4,J) = CODE TO DEFINE THETA 3 ROTATION CONSTRAINT
C               IHDATA(5,J) = CODE TO DEFINE X TRANSLATION CONSTRAINT
C               IHDATA(6,J) = CODE TO DEFINE Y TRANSLATION CONSTRAINT
C               IHDATA(7,J) = CODE TO DEFINE Z TRANSLATION CONSTRAINT
C               CONSTRAINT CODE = 0 - NO CONSTRAINT
C               CONSTRAINT CODE = 1 - FIXED CONSTRAINT, NO RELATIVE
C                                     MOTION ALLOWED IN THIS DIRECTION
C               CONSTRAINT CODE = 2 - RHEONOMIC CONSTRAINT, RELATIVE
C                                     MOTION IN THIS DIRECTION TO BE
C                                     USER DEFINED VIA SUBROUTINES
C                                     ADT AND ADDT
C
C        LINEAR OR NON-LINEAR ANALYSIS
C        -----------------------------
C     IFLNER = INPUT PARAMETER IPDATA(3) WHICH IS READ IN, IN DYNSIO
C              AND STORED IN /ILINER/
C              IFLNER = 0 IF THE USER DESIRES THE COUPLED NON-LINEAR
C                         EQUATIONS OF MOTION TO BE INTEGRATED WITH
C                         RESULTANT TIME RESPONSE OUTPUT
C              IFLNER = 1 IF THE USER DESIRES THE COUPLED NON-LINEAR
C                         EQUATIONS OF MOTION TO BE NUMERICALLY
C                         LINEARIZED AND EITHER THE LINEAR TIME OR
C                         FREQUENCY RESPONSE ANALYSIS PERFORMED
C
C        STABILITY ANALYSIS:   (TRANSFER FUNCTIONS MAY BE OBTAINED
C        ------------------     WHICH DEFINE THE FREQUENCY RESPONSE
C                               CHARACTERISTICS OF ANY USER DEFINED
C                               OUTPUT 'TORQUE SIGNAL' TO ANY USER
C                               DEFINED INPUT 'SENSOR SIGNAL')
C
C     NXSS   = TOTAL NUMBER OF USER DEFINED SENSOR SIGNALS. THESE ARE
C              ASSUMED TO BE FUNCTIONS OF THE PLANT STATE VARIABLES AND
C              ARE NUMERICALLY LINEARIZED WITH RESPECT TO THEM. THIS
C              PARAMETER IS USER DEFINED IN CONTRL AND STORED
C              IN /LOSIZE/
C     NBTQ   = TOTAL NUMBER OF USER DEFINED TORQUE SIGNALS. THESE ARE
C              ASSUMED TO BE FUNCTIONS OF THE CONTROL SYSTEM STATE
C              VARIABLES AND ARE NUMERICALLY LINEARIZED WITH RESPECT TO
C              THEM. THIS PARAMETER IS USER DEFINED IN CONTRL AND STORED
C              IN /LOSIZE/
C     NAUX   = TOTAL NUMBER OF AUXILARY EQUATIONS, DEFINED IN EQADD
```

```
C              AND STORED IN /DNAUX/                                      017466
C                             NAUX = NXSS + NBTQ                          017467
C      NDELTA  = DEFINED EQUAL TO NDELTA IN CENTRL, STORED IN /LDSIZE/    017468
C                                                                         017469
C                                                                         017470
C                                                                         017471
C     DISCOS COMPUTATION CONTROL PARAMETERS                               017472
C     -------------------------------------                               017473
C                                                                         017474
C          STATE VECTOR, SIZE AND ARRANGEMENT OF STATE VARIABLES          017475
C          -----------------------------------------------------          017476
C     THE FOLLOWING PARAMETERS ARE INTEGER*4 CONSTANTS, COMPUTED          017477
C       IN DYNSIO AND STORED IN /SPECIF/. THEY ARE USED TO DEFINE THE     017478
C       NUMBER OF AND LOCATION OF VARIOUS COMPONENTS OF THE TOTAL         017479
C       SYSTEM STATE VECTOR                                               017480
C     NBETA  = TOTAL NUMBER OF (BETA) COORDINATES IN THE STATE VECTOR.    017481
C              THESE COORDINATES DEFINE THE TRANSLATION OR ROTATIONAL     017482
C              MOTION ASSOCIATE WITH EACH UNCONSTRAINED OR RHENOMICALLY   017483
C              CONSTRAINED DEGREE OF FREEDOM                              017484
C     NLAM   = TOTAL NUMBER OF (LAMBA) CONSTRAINT EQUATIONS. A (LAMBA)    017485
C              CONSTRAINT EQUATION IS DEFINED FOR EACH FIXED OR           017486
C              RHEONOMICALLY CONSTRAINED DEGREE OF FREEDOM                017487
C     NU     = TOTAL NUMBER OF (U'S) VELOCITY COORDINATES IN THE STATE    017488
C              VECTOR. THE BREAK-DOWN OF VELOCITY COORDINATES FOR BODY N  017489
C              IS AS FOLLOWS:                                             017490
C                3 - ROTATION COORDINATES (COMPONENTS OF THE INERTIAL     017491
C                    ANGULAR VELOCITY VECTOR WRT THE BODY N REFERENCE)    017492
C                3 - TRANSLATION COORDINATES (COMPONENTS OF THE INERTIAL  017493
C                    LINEAR VELOCITY VECTOR WRT THE BODY N REFERENCE)     017494
C        IRGFLX(N) - GENERALIZED MODAL VELOCITY COORDINATES ASSOCIATED    017495
C                    WITH FLEXIBLE BODY N DEFORMATION, ZERO IF BODY N     017496
C                    IS A RIGID BODY                                      017497
C        NNEW(2,N) - WHEEL SPIN RATE COORDINATES ASSOCIATED WITH THE      017498
C                    ACTIVE OR VARIABLE SPEED WHEELS IN BODY N            017499
C                    ZERO IF BODY N HAS NO EMBEDDED ACTIVE WHEELS         017500
C     NEQ    = TOTAL NUMBER OF STATE EQUATIONS WHICH ARE SETUP TO         017501
C              DEFINE TOTAL SYSTEM TIME RESPONSE. IN DYNSIO               017502
C                                                                         017503
C                NEQ = IRGFLX(1) + ... + IRGFLX(NB)                       017504
C                          + NU + NBETA + NDELTA                          017505
C                                                                         017506
C     LENU   = STATE VECTOR LENGTH TABLE, COMPUTED INTEGER*4 ARRAY        017507
C              DIMENSIONED (2*NBMAX+2)                                    017508
C                LENU(N) = NUMBER OF VELOCITY COORDINATES (U'S)           017509
C                          ASSOCIATED WITH BODY N, N=1,2,...,NB           017510
C             LENU(N+NB) = NUMBER OF MODAL DISPLACEMENT COORDINATES       017511
C                          ASSOCIATED WITH BODY N, N=1,2,...,NB           017512
C           LENU(2*NB+1) = NBETA                                          017513
C           LENU(2*NB+2) = NDELTA                                         017514
C     LOCU   = STATE VECTOR LOCATION TABLE, COMPUTED INTEGER*4 ARRAY      017515
C              DIMENSIONED (2*NBMAX+2)                                    017516
C                LOCU(N) = LOCATION IN STATE VECTOR OF FIRST VELOCITY     017517
C                          COORDINATE ASSOCIATED WITH BODY N, N=1,...,NB  017518
C             LOCU(N+NB) = LOCATION IN STATE VECTOR OF FIRST MODAL        017519
C                          DISPLACEMENT COORDINATE ASSOCIATED WITH        017520
C                          BODY N, N=1,2,...,NB                           017521
C           LOCU(2*NB+1) = LOCATION IN STATE VECTOR OF FIRST (BETA)       017522
C                          COORDINATE USED TO DEFINE CONTIGUOUS BODY      017523
C                          RELATIVE MOTION                                017524
C           LOCU(2*NB+2) = LOCATION IN STATE VECTOR OF FIRST STATE        017525
```

```
C                         VARIABLE ASSOCIATED WITH THE NDELTA USER      017526
C                         DEFINED CONTROL EQUATIONS                     017527
C                                                                       017528
C     THE FOLLOWING PARAMETERS ARE INTEGER*4 CONSTANTS, COMPUTED        017529
C      INTERNALLY AND STORED IN /ZDSIZE/. THEY ARE USED IN THE SETUP    017530
C      OF THE LINEARIZED EQUATIONS OF MOTION                            017531
C     NX      = NUMBER OF INDEPENDENT COORDINATES IN THE STATE VECTOR Y. 017532
C               THIS IS COMPUTED IN LINEAR                              017533
C     NJC     = NUMBER OF INDEPENDENT COORDINATES IN THE STATE VECTOR Y  017534
C               PLUS THE NUMBER OF USER DEFINED SENSOR AND TORQUE SIGNALS 017535
C               (AUXILARY EQUATIONS) CODED IN EQADD; COMPUTED IN LINEAR 017536
C                          NJC = NX + NAUX                              017537
C     NY2     = NUMBER OF INDEPENDENT PLANT COORDINANTS IN THE STATE    017538
C               VECTOR MINUS THE NUMBER OF USER DEFINED SENSOR SIGNALS  017539
C                          NY2 = NX - NDELTA - NXSS                     017540
C               COMPUTED IN DYNS40                                      017541
C     ND2     = NUMBER OF INDEPENDENT CONTROL SYSTEM COORDINATES IN THE 017542
C               STATE VECTOR MINUS THE NUMBER OF USER DEFINED TORQUE    017543
C               SIGNALS                                                 017544
C                          ND2 = NDELTA - NBTQ                          017545
C               COMPUTED IN DYNS40                                      017546
C                                                                       017547
C         INTEGRATION CONTROL                                           017548
C         -------------------                                           017549
C     JIL     = 1,2,3 OR 4; DEFINES THE CYCLE NUMBER WITHIN THE FOUR STEP 017550
C               RUNGE KUTTA INTEGRATION ROUTINE THAT COMPUTATION IS     017551
C               CURRENTLY AT; UPDATED IN RKADAM AND STORED IN /JILFLG/. 017552
C       NOTE    ADAMS PREDICTOR/CORRECTOR INTEGRATION MAY BE USED BY    017553
C       ----    SETTING ITYPE=2 IN THE DATA STATEMENT IN RKADAM. IF SO  017554
C               JIL WILL ALWAYS BE EQUAL TO 4 AFTER THE FIRST RUNGE     017555
C               KUTTA STARTING STEP.                                    017556
C                                                                       017557
C     THE FOLLOWING ARE REAL*8 PARAMETERS STORED IN /TIMESS/ AND USED   017558
C      TO DEFINE DIFFERENT TIME PARAMETERS                              017559
C     ST      = STARTT = INPUT PARAMETER TMDATA(1) READ IN, IN DYNS10;  017560
C                        USED TO DEFINE THE STARTING TIME FOR TIME      017561
C                        HISTORY SIMULATION                             017562
C     DT      = DELTAT = INPUT PARAMETER TMDATA(2) READ IN, IN DYNS10;  017563
C                        USED TO DEFINE THE INTEGRATION STEP SIZE       017564
C     ET      = ENDT   = INPUT PARAMETER TMDATA(3) READ IN, IN DYNS10;  017565
C                        USED TO DEFINE THE END OF SIMULATION TIME      017566
C     T       = TIME PARAMETER, INCREMENTED IN RKADAM. COMPUTATION      017567
C               STARTS AT T=ST AND IS STOPPED AT T=ET                   017568
C     TMST    = TIME ELAPSED IN SIMULATION FROM T=ST                    017569
C                                                                       017570
C     THE FOLLOWING ARE VARIABLES INITIALIZED IN DYNS20, UPDATED IN     017571
C      RKADAM AND STORED IN /GPRKTA/                                    017572
C     GRK    = REAL*8 WORK AREA FOR RUNGE-KUTTA INTEGRATION ROUTINE     017573
C              DIMENSIONED (KY)                                         017574
C     FRK    = REAL*8 ARRAY OF CONSTANTS REQUIRED FOR RUNGE KUTTA       017575
C              INTEGRATION ROUTINE DIMENSIONED (4)                      017576
C     NT     = INTEGER*4 VARIABLE USED AS A COUNTER. ITS STORED VALUE IS 017577
C              EQUAL TO THE TOTAL NUMBER OF INTEGRATION STEPS TAKEN IN  017578
C              NON-LINEAR TIME RESPONSE ANALYSIS FROM TIME STARTT TO    017579
C              INSTANTANEOUS TIME T.                                    017580
C                                                                       017581
C         OUTPUT DATA CONTROL                                           017582
C         -------------------                                           017583
C     NJPRNT = INPUT PARAMETER IPDATA(1) READ IN, IN DYNS10 AND STORED  017584
C               IN /MISCNO/. PRINT OUTPUT DATA EVERY NJPRNT MULTIPLES   017585
```

```
C            OF THE INTEGRATION STEP
C     NOPLOT = INPUT PARAMETER IPDATA(2) READ IN, IN DYSNIO AND STORED
C            IN /MISCNO/. PLOT OUTPUT DATA EVERY NOPLOT MULTIPLES
C            OF THE INTEGRATION STEP
C     NRPLOT = BOTH ARE INTEGERS COMPUTED IN PLOTWR, LPLTWR AND DYNS30
C     NCPLOT   AND STORED IN /PLTDTA/ THEY ARE REQUIRED FOR SC-4020
C            PLOT GENERATION
C     KSAVE  = TOTAL NUMBER OF DATA POINTS TO BE SAVED IN THE /PSTUFF/
C            ARRAYS, WHICH ARE EXCLUSIVELY RESERVED FOR THE STORAGE OF
C            FREQUENCY RESPONSE DATA COMPUTED IN SFREQZ
C
C
C     MODE DEPENDENT PARAMETERS
C-----------------------------
C
C         MASS MATRIX WITH ASSOCIATED MODE DEPENDENT PARAMETERS
C         -----------------------------------------------------
C
C     AMU    = MASS MATRIX ARRAY, REAL*8 TIME VARYING PARAMETERS STORED
C            IN /AMUBA/, DIMENSIONED(KMU,KMU,N3MAX)
C             UPON ENTRY INTO DYNS20:
C            ((AMU(I,J,N),I=1,LV),J=1,LV) = MASS MATRIX FOR BODY N
C                                   IN ITS UNDEFORMED STATE WHERE
C                                     LV = 6 + IRGFLX(N) + NMOW(2,N)
C             THEREAFTER:
C            ((AMU(I,J,N),I=1,LV),J=1,LV) =MASS MATRIX FOR BODY N
C                                   AT TIME T IN THE DEFOMRED STATE
C                                   (COMPUTED IN MGEN), WHERE
C                                     LV = 6 + IRGFLX(N) + NMOW(2,N)
C            DETAILS SEE EQUATIONS II-39 AND 55 IN VOL I:
C        ((AMU(I,J,N),I=1,3),J=1,3)    = INERTIA TENSOR FOR BODY N
C                                        RELATIVE TO BODY N REF POINT WRT
C                                        BODY N REFERENCE FRAME
C        ((AMU(I,J,N),I=1,3),J=4,6)    = SKEW TENSOR FORM OF THE BODY N
C                                        MASS MOMENT VECTOR RELATIVE TO
C                                        BODY N REF POINT WRT BODY N
C                                        REFERENCE FRAME
C        ((AMU(I,J,N),I=4,6),J=4,6)    = DIAGONAL MASS MATRIX, TOTAL MASS
C                                        OF BODY N ON DIAGONAL, ZERO OFF
C                                        THE DIAGONAL
C               FOR THE FOLLOWING LET:
C                 LE = 6+IRGFLX(N)
C                 LV = 6+IRGFLX(N)+NMOW(2,N)
C        ((AMU(I,J,N),I=1,3),J=7,LE)   = INERTIAL COUPLING TERMS BETWEEN
C                                        ANGULAR AND MODAL ACCELERATION
C                                        COORDINATES FOR BODY N
C        ((AMU(I,J,N),I=4,6),J=7,LE)   = INERTIAL COUPLING TERMS BETWEEN
C                                        LINEAR AND MODAL ACCELERATION
C                                        COORDINATES FOR BODY N
C        ((AMU(I,J,N),I=7,LE),J=7,LE) = INERTIAL COUPLING TERMS BETWEEN
C                                        MODAL ACCELERATION COORDINATES
C                                        FOR BODY N (DIAGONAL FOR
C                                        ORTHOGONAL MODES)
C        ((AMU(I,J,N),I=1,LE),J=LE+1,LV)     = INERTIAL COUPLING TERMS
C                                              BETWEEN BODY N ACCEL.
C                                              COORD. AND WHEEL ACCEL.
C        ((AMU(I,J,N),I=LE+1,LV),J=LE+1,LV) = INERTIAL COUPLING BETWEEN
C                                              VARIABLE SPEED WHEELS ON
C                                              BODY N
C             IN SUBROUTINE YDOT THE MASS MATRIX INVERSE IS NEEDED. IT
```

```
C             IS COMPUTED IN INVINP AND STORED IN THE AMU ARRAY           017646
C             THUS DESTROYING ALL MASS MATRIX DATA DEFINED BELOW          017647
C             ** NOTE IT MAY BE RECOVERED BY REINVERTING AMU ARRAY        017648
C                                                                         017649
C     THE FOLLOWING PARAMETERS ARE REAL*8 ARRAYS STORED IN /INTGRL/.      017650
C        THEIR ELEMENT MAGNITUDES ARE EVALUATED IN EITHER MRIGID, MSMODC  017651
C        OR MSMODE AND STORED ON NTAPE1 (TO CONSERVE COMPUTE STORAGE).    017652
C        UPON ENTRY INTO DYNS20 A READ(NTAPE1) INSTRUCTION IS EXECUTED    017653
C        AND THE APPROPRIATE TIME INVARIANT ARRAYS LOADED.                017654
C        THESE ARRAY WILL BE USED IN MGEN TO COMPUTE THE ELEMENTS OF THE  017655
C        TIME VARYING MASS MATRIX DEFINED BY EQUATION II-86, VOL 1        017656
C     AM    = UNDEFORMED MASS MATRIX ARRAY (M SUB J IN II-86)             017657
C             DIMENSIONED (((6+NMODBOD)*(7+NMODBOD))/2,NBMAX). UPON ENTRY 017658
C             INTO DYNS20 IT IS SET BY THE FOLLOWING CODE:                017659
C                    DO 3 N=1,NB                                          017660
C                    NAL = IRGFLX(N)                                      017661
C                    NP6 = 6+NAL                                          017662
C                    KNT = 0                                              017663
C                    DO 6 I=1,NP6                                         017664
C                    DO 6 J=1,NP6                                         017665
C                    KNT = KNT+1                                          017666
C                  6 AM(KNT,N) = AMU(I,J,N)                               017667
C                  3 CONTINUE                                             017668
C             SINCE THE MASS MATRIX IS SYMMETRIC                          017669
C             ONLY THE UPPER TRIANGULAR PORTION NEED BE SAVED             017670
C                 (AM(I,N),I=1,(NP6*(NP6+1))/2) = UNDEFORMED MASS         017671
C                                                  MATRIX FOR BODY N      017672
C     ACOF  = COEFFICIENTS WHICH DEFINE THE LINEAR DEPENDENCE OF THE      017673
C             MASS MOMENT ON FLEXURAL DEFORMATION,                        017674
C             DIMENSIONED (9,NMODBOD,NBMAX). THAT IS:                     017675
C             (ACOF(I,M,N),I=1,9) = 9 ALPHA COEFFICIENTS IN EQ II-88      017676
C                                    WHICH DEFINED THE LINEAR DEPENDENCE  017677
C                                    OF BODY N MASS MOMENT ON BODY N      017678
C                                    MODE M DEFORMATION.                  017679
C     BCOF  = COEFFICIENTS WHICH DEFINE THE LINEAR DEPENDENCE OF THE      017680
C             INERTIA TENSOR ON FLEXURAL DEFORMATION,                     017681
C             DIMENSIONED (6,NMODBOD,NBMAX). THAT IS:                     017682
C             (BCOF(I,M,N),I=1,6) = 6 LOWER CASE B COEFFICIENTS IN        017683
C                                    EQ II-88 WHICH DEFINE THE LINEAR     017684
C                                    DEPENDENCE OF BODY N INERTIA TENSOR  017685
C                                    ON BODY N MODE M DEFORMATION         017686
C     COFYZ = COEFFICIENTS WHICH DEFINE THE LINEAR DEPENDENCE OF THE      017687
C     COFXZ   INERTIAL COUPLING TERMS BETWEEN ANGULAR AND MODAL           017688
C     COFXY   ACCELERATION COEFFICIENTS, EACH OF THE THREE ARRAYS ARE     017689
C             DIMENSIONED (NMODBOD,NMODBOD,NBMAX)                         017690
C             (COFYZ(I,M,N),I=1,IRGFLX(N))= IRGFLX(N) C SUB YZ COUPLING   017691
C                                    COEFFICIENTS IN EQ II-88. DEFINES    017692
C                                    THE LINEAR DEPENDENCE OF BODY N      017693
C                                    X-AXIS ROTATION - MODE M COUPLING    017694
C                                    COEFFICIENTS ON MODE I,I=1,.....     017695
C                                    IRGFLX(N) DEFORMATION                017696
C             (COFXZ(I,M,N),I=1,IRGFLX(N))= IRGFLX(N) C SUB XZ COUPLING   017697
C                                    COEFFICIENTS IN EQ II-88. COUPLING   017698
C                                    Y-AXIS ROTATION - MODE M BODY N      017699
C             (COFXY(I,M,N),I=1,IRGFLX(N))= IRGFLX(N) C SUB XY COUPLING   017700
C                                    COEFFICIENTS IN EQ II-88. COUPLING   017701
C                                    Z-AXIS ROTATION - MODE M BODY N      017702
C     COF11 = COEFFICIENTS WHICH DEFINE THE QUADRATIC DEPENDENCE OF THE   017703
C     COF12   INERTIA TENSOR ON FLEXURAL DEFORMATION,                     017704
C     COF13   DIMENSIONED (NMODBOD,NMODBOD,NBMAX)                         017705
```

```
C       [illegible]   COEF11(I,J,N) = COEFFICIENTS WHICH DEFINE THE QUADRATIC    017706
C       [illegible]   COEF22(I,J,N)   DEPENDENCE OF PRINCIPAL XX, YY, ZZ MOMENTS  017707
C       [illegible]   COEF33(I,J,N)   OF INERTIA RESP. OF BODY N ON MODE I AND    017708
C                                     MODE J FLEXURAL DEFORMATION. SEE EQ II-89   017709
C                     COEF12(I,J,N) = COEFFICIENTS WHICH DEFINE THE QUADRATIC    017710
C                     COEF13(I,J,N)   DEPENDENCE OF PRODUCTS XY, XZ, YZ           017711
C                     COEF23(I,J,N)   OF INERTIA RESP. OF BODY N ON MODE I AND    017712
C                                     MODE J FLEXURAL DEFORMATION. SEE EQ II-89   017713
C                                                                                 017714
C         MODAL COMPONENTS AT HINGE AND SENSOR POINTS                             017715
C         - -------------------------------------------                           017716
C     THE FOLLOWING PARAMETERS ARE REAL*8 ARRAYS STORED IN /HANDS/ THE            017717
C        ACTUAL DATA TO BE LOADED INTO THESE ARRAYS IS INPUTTED EITHER            017718
C        VIA MSMODL OR MSMODC. IT IS STORED TEMPORARILY ON NTAPE1 THEN            017719
C       UNLOADED IN DYNS20 INTO THE APPROPRIATE ARRAYS                            017720
C     FAIH    = HINGE POINT MODAL DISPLACEMENT ARRAY,                             017721
C             DIMENSIONED (3,NMODBD,2*NHMAX-1),                                   017722
C     SIGH    = HINGE POINT MODAL ROTATION ARRAY,                                 017723
C             DIMENSIONED (3,NMODBD,2*NHMAX-1)                                    017724
C               AT HINGE POINT NOH, NOH=2,3,...,NH                                017725
C             THE G-FRAME IS CONTAINED IN BODY N=ITOPOL(1,NOH) AND                017726
C             THE P-FRAME IS CONTAINED IN BODY M=ITOPOL(2,NOH)                    017727
C        (FAIH(I,NN,2*NOH-3),I=1,3) = 3 DISPLACEMENT COMPONENTS OF MODE           017728
C                                     NN AT THE G-FRAME ORIGIN WRT THE            017729
C                                     BODY N FIXED REFERENCE                      017730
C        (FAIH(I,MM,2*NOH-2),I=1,3) = 3 DISPLACEMENT COMPONENTS OF MODE           017731
C                                     MM AT THE P-FRAME ORIGIN WRT THE            017732
C                                     BODY M FIXED REFERENCE                      017733
C        (SIGH(I,NN,2*NOH-3),I=1,3) = 3 ROTATION COMPONENTS OF MODE               017734
C                                     NN AT THE G-FRAME ORIGIN WRT THE            017735
C                                     BODY N FIXED REFERENCE                      017736
C        (SIGH(I,MM,2*NOH-2),I=1,3) = 3 ROTATION COMPONENTS OF MODE               017737
C                                     MM AT THE P-FRAME ORIGIN WRT THE            017738
C                                     BODY M FIXED REFERENCE                      017739
C     FAIS    = SENSOR POINT MODAL DISPLACEMENT ARRAY,                            017740
C             DIMENSIONED (3,NMODBD,NSPMAX)                                       017741
C     SIGS    = SENSOR POINT MODAL ROTATION ARRAY                                 017742
C             DIMENSIONED (3,NMODBD,NSPMAX)                                       017743
C               AT SENSOR POINT NOS WHICH IS ON BODY N=IFTSMW(NOS)                017744
C        (FAIS(I,NN,NOS),I=1,3) = 3 DISPLACEMENT COMPONENTS OF MODE NN            017745
C                                 AT SENSOR POINT NOS WRT BODY N REF              017746
C        (SIGS(I,NN,NOS),I=1,3) = 3 ROTATION COMPONENTS OF MODE NN                017747
C                                 AT SENSOR POINT NOS WRT BODY N REF              017748
C                                                                                 017749
C         MODAL STIFFNESS AND DAMPING MATRICES                                    017750
C         ------------------------------------                                    017751
C                                                                                 017752
C     THE FOLLOWING PARAMETERS ARE REAL*8 ARRAYS STORED IN /INTGRL/.              017753
C        THESE ARRAYS ARE LOADED IN DYNS20 BY A READ(NTAPE1) STATEMENT.           017754
C        THE DATA ON NTAPE1 WAS GENERATED IN AND BY CALLS FROM THE                017755
C        FLEXIBLE BODY INPUT ROUTINES MSMODL AND MSMODC                           017756
C     AK      = MODAL STIFFNESS MATRIX, DIMENSIONED (NMODBD,NMODBD,NBMAX)         017757
C             LET:                                                                017758
C                 NM =IRGFLX(N) = NUMBER OF MODES USED FOR BODY N                 017759
C             THEN                                                                017760
C            ((AK(I,J,N),I=1,NM),J=1,NM) = NM BY NM MODAL STIFFNESS               017761
C                                 MATRIX FOR BODY N. NON-ORTHOGONAL MODES         017762
C                                 MAY BE USED; IF SO, THIS MATRIX WILL BE         017763
C                                 SYMMETRIC BUT NON-DIAGONAL.                     017764
C             FOR ORTHOGONAL MODES:                                               017765
```

```
C           AK(M,M,N) = (GENERALIZED MASS OF MODE M BODY N) * (MODE M   017766
C                             NATURAL FREQUENCY)**2                     017767
C     AD      = MODAL DAMPING MATRIX, DIMENSIONED (NMODBD,NMODBD,NBMAX) 017768
C           LET:                                                        017769
C              NM = IROFLX(N) = NUMBER OF MODES USED FOR BODY N         017770
C           THEN                                                        017771
C          ((AD(I,J,N),I=1,NM),J=1,NM) = NM BY NM MODAL DAMPING         017772
C                             MATRIX FOR BODY N. IF THE DAMPING MATRIX  017773
C                             IS NOT PROPORTIONAL TO A LINEAR FUNCTION  017774
C                             OF THE MASS AND STIFFNESS MATRICES THIS   017775
C                             MODAL DAMPING MATRIX WILL BE SYMMETRIC AND 017776
C                             NON-DIAGONAL                              017777
C           FOR PROPORTIONAL DAMPING IN BODY N:                         017778
C           AD(M,M,N) = 2 * (GENERALIZED MASS OF MODE M)                017779
C                         * (DAMPING ZETA FOR MODE M)                   017780
C                         * (NATURAL FREQUENCY OF MODE M)               017781
C                                                                       017782
C                                                                       017783
C     GRAVITY GRADIENT DATA, INPUTTED AND COMPUTED                      017784
C----------------------------------------------------                   017785
C                                                                       017786
C     GV      = GRAVITY GRADIENT INPUT DATA ARRAY. THESE REAL*8 CONSTANTS 017787
C           ARE USED TO COMPUTE DATA TO BE STORED IN /GGDATA/           017788
C           (GV(I),I=1,3) = 3 COMPONENTS OF THE GRAVITY VECTOR WRT      017789
C                              INERTIAL REFERENCE (UNITS OF ACC)        017790
C           GV(4)         = DISTANCE FROM GRAVITY SOURCE TO VICINITY    017791
C                              OF BODY CLUSTER CENTER OF MASS           017792
C                                                                       017793
C     THE FOLLOWING PARAMETERS ARE REAL*8 VARIABLES STORED IN /GGDATA/  017794
C        THE USER MAY, IN CONTROL, DEFINE THESE PARAMETERS TO BE TIME   017795
C        FUNCTIONS SO THAT MOTION IN ORBIT MAY BE SIMULATED             017796
C     RCVAC  = GV(4) DISTANCE FROM GRAVITY SOURCE TO VICINITY OF SYSTEM 017797
C     GMAG   = RESULTANT GRAVITATIONAL ACCELERATION                     017798
C                GMAG = DSQRT(GV(1)**2 + GV(2)**2 + GV(3)**2)           017799
C     GAMGI  = COMPONENTS OF A UNIT VECTOR FROM THE GRAVITY SOURCE TO   017800
C           THE VICINITY OF THE SYSTEM CENTER OF MASS WRT INERTIAL      017801
C           REFERENCE, DIMENSIONED (3). FROM DYNSIO                     017802
C            GAMGI(I) = GV(I)/GMAG; I=1,2,3                             017803
C             (NOT COMPUTE IF (GMAG.EQ.0))                              017804
C     GGS     = GRAVITY GRADIENT EFFECTS ON ELASTIC COORDINATES, A REAL*8 017805
C           ARRAY DIMENSIONED (NMODBD,9,NBMAX) AND STORED IN /GCSAVE/   017806
C           THE ELEMENTS OF THIS ARRAY ARE COMPUTED IN MGEN BASED       017807
C           UPON THEORY ON PAGES 48-52, VOL I.  THEY ARE                017808
C           DEFORMATION DEPENDENT TERMS REQUIRED FOR COMPUTATION        017809
C           OF GRAVITY GRADIENT FORCES AND TORQUES                      017810
C                FOR BODY N LET                                         017811
C                   LO = LOCQ(NB+N) = LOCATION OF FIRST ELASTIC COORD.  017812
C                                       IN STATE VECTOR                 017813
C                   LE = LENQ(NE+N) = NUMBER OF ELASTIC COORD. IN Y     017814
C        (GGS(J,1,N),J=1,LE) = ROW J OF COF11(1,1,N)*Y(LO)              017815
C        (GGS(J,2,N),J=1,LE) = ROW J OF COF22(1,1,N)*Y(LO)              017816
C        (GGS(J,3,N),J=1,LE) = ROW J OF COF33(1,1,N)*Y(LO)              017817
C        (GGS(J,4,N),J=1,LE) = ROW J OF COF12(1,1,N)*Y(LO)              017818
C        (GGS(J,5,N),J=1,LE) = ROW J OF COF13(1,1,N)*Y(LO)              017819
C        (GGS(J,6,N),J=1,LE) = ROW J OF COF23(1,1,N)*Y(LO)              017820
C        (GGS(J,7,N),J=1,LE) = COL J OF Y(LO)**T*COF12(1,1,N)           017821
C        (GGS(J,8,N),J=1,LE) = COL J OF Y(LO)**T*COF13(1,1,N)           017822
C        (GGS(J,9,N),J=1,LE) = COL J OF Y(LO)**T*COF23(1,1,N)           017823
C                                                                       017824
C                                                                       017825
```

```
C      KINEMATIC DATA INPUTTED AND COMPUTED
C----------------------------------------------
C
C      BETAH  = CONTIGUOUS HINGE FRAME ORIENTATION DATA ARRAY, REAL*8
C               PARAMETERS STORED IN /SPECIF/, DIMENSIONED (6,NHMAX)
C               INITIAL VALUES INPUTTED IN DYNSIO
C               INSTANTANEOUS VALUES ARE STRIPPED FROM THE STATE VECTOR
C               ARRAY Y IN ROTOH
C               (BETAH(I,N),I=1,3) = 3 SUCCESSIVE EULER ANGLES REQUIRED
C                                    TO ORIENT Q-FRAME RELATIVE TO
C                                    P-FRAME AT HINGE POINT N
C               (BETAH(I,N),I=4,6) = 3 POSITION COORDINATES OF THE ORIGIN
C                                    OF THE Q-FRAME RELATIVE TO THE
C                                    P-FRAME AT HINGE POINT N
C      BETAHD = CONTIGUOUS HINGE FRAME RELATIVE RATE DATA ARRAY, REAL*8
C               PARAMETERS STORED IN /SPECIF/, DIMENSIONED (6,NHMAX)
C               INITIAL VALUES INPUTTED IN DYNSIO
C               INSTANTANEOUS VALUES ARE STRIPPED FROM THE STATE VECTOR
C               OUT ARRAY YDT IN YDOT
C                 BETAHD(I,N) = FIRST TIME DERIVATIVE OF BETAH(I,N)
C                               ((I=1,6),N=1,NH)
C      AMO      = MOMENTUM WHEEL DATA ARRAY, REAL*8 PARAMETERS STORED IN
C               /SPECIF/, DIMENSIONED (2,NMMAX), INPUT DATA FROM DYNSIO
C               THESE PARAMETERS ARE NOT FUNCTIONS OF TIME
C                    AMO(1,M) = INITIAL SPIN RATE FOR WHEEL M
C                    AMO(2,M) = SPIN INERTIA FOR WHEEL M
C
C
C        ROTOH - TRANSFORMATION MATRICES COMPUTED THEREIN
C        ----------------------------------------------
C      THE FOLLOWING ARRAYS ARE FUNCTIONS OF TIME. ELEMENTS OF THE STATE
C         VECTOR Y ARE USED IN ROTOH TO COMPUTE THE KINEMATIC
C         TRANSFORMATION MATRICES ASSOCIATED WITH HINGE POINT AND BODY
C         FIXED REFERENCE FRAMES, SET DEBUG(44) = .TRUE. TO PRINT ARRAYS
C
C      BETAH  = EULER ANGLES UPDATED FROM STATE VECTOR Y AT TIME T IN
C               SUBROUTINE ROTOH
C      OH       = REFERENCE POINT, HINGE POINT POSITION VECTOR ARRAY
C               DIMENSIONED (3,7*(NHMAX-1)), STORED IN /SPECIF/. ACROSS
C               HINGE POINT NOH, NOH=2,3,....,NH THE FOLLOWING POSITION
C               VECTORS ARE COMPUTED. LET
C                         LOI = 7*(NOH-2) + 1
C               THEN
C           (OH(J,LOI+0),J=1,3) = 3 COORDINATES OF THE ORIGIN OF THE
C                                 Q-FRAME ON BODY N=ITOPOL(1,NOH) WRT THE
C                                 BODY N REFERENCE IN THE UNDEFORMED
C                                 STATE (MSMODE OR MSMODE)
C           (OH(J,LOI+1),J=1,3) = 3 COORDINATES OF THE ORIGIN OF THE
C                                 P-FRAME ON BODY M=ITOPOL(2,NOH) WRT THE
C                                 BODY M REFERENCE IN THE UNDEFORMED
C                                 STATE (MSMODE OR MSMODE)
C           (OH(J,LOI+2),J=1,3) = 3 COORDINATES OF THE ORIGIN OF THE
C                                 Q-FRAME ON BODY N=ITOPOL(1,NOH) WRT THE
C                                 BODY N REFERENCE, DEFORMED STATE
C           (OH(J,LOI+3),J=1,3) = 3 COORDINATES OF THE ORIGIN OF THE
C                                 P-FRAME ON BODY M=ITOPOL(2,NOH) WRT THE
C                                 BODY M REFERENCE DEFORMED STATE
C           (OH(J,LOI+4),J=1,3) = 3 COORDINATES OF THE ORIGIN OF THE
C                                 Q-FRAME WRT THE P-FRAME AT HINGE
C                                 POINT NOH
```

```
C         (OH(J,LOI+5),J=1,3) = 3 COORDINATES OF THE ORIGIN OF THE        017886
C                               BODY N=ITOPOL(1,NOH) FIXED REFERENCE      017887
C                               WRT THE BODY M=ITOPOL(2,NOH) FIXED        017888
C                               REFERENCE                                 017889
C         (OH(J,LOI+6),J=1,3) = 3 COORDINATES OF THE POSITION VECTOR      017890
C                               FROM THE ORIGIN OF THE BODY               017891
C                               N=ITOPOL(1,NOH) FIXED REFERENCE TO THE    017892
C                               ORIGIN OF THE BODY M=ITOPOL(2,NOH)        017893
C                               FIXED REFERENCE WRT THE INERTIAL          017894
C                               REFERENCE FRAME                           017895
C     RH      = BODY FIXED REFERENCE FRAME, HINGE POINT REFERENCE FRAME   017896
C               TRANSFORMATION MATRICES, DIMENSIONED (3,3,6*(NHMAX-1)),   017897
C               STORED IN /SPECIF/. ACROSS HINGE POINT NOH, NOH=2,...,NH  017898
C               THE FOLLOWING COORDINATE TRANSFORMATION MATRICES ARE      017899
C               COMPUTED. LET                                             017900
C                         LRI = 6*(NOH-2) + 1                             017901
C               THEN                                                      017902
C        ((RH(I,J,LRI+0),I=1,3),J=1,3) = 9 ELEMENTS OF TRANSFORMATION     017903
C                               MATRIX USED TO TAKE VECTORS FROM          017904
C                               Q-FRAME COORDINATES AT HINGE POINT NOH    017905
C                               TO BODY N=ITOPOL(1,NOH) FIXED REFERENCE   017906
C                               FRAME COORDINATES IN BODY N UNDEFORMED    017907
C                               STATE                                     017908
C        ((RH(I,J,LRI+1),I=1,3),J=1,3) = 9 ELEMENTS OF TRANSFORMATION     017909
C                               MATRIX USED TO TAKE VECTORS FROM          017910
C                               P-FRAME COORDINATES AT HINGE POINT NOH    017911
C                               TO BODY M=ITOPOL(2,NOH) FIXED REFERENCE   017912
C                               FRAME COORDINATES IN BODY M UNDEFORMED    017913
C                               STATE                                     017914
C        ((RH(I,J,LRI+2),I=1,3),J=1,3) = TRANSFORMATION MATRIX Q-FRAME    017915
C                               COORDINATES AT HINGE POINT NOH TO         017916
C                               BODY N=ITOPOL(1,NOH) FIXED COORDINATES    017917
C                               IN BODY N DEFORMED STATE                  017918
C        ((RH(I,J,LRI+3),I=1,3),J=1,3) = TRANSFORMATION MATRIX P-FRAME    017919
C                               COORDINATES AT HINGE POINT NOH TO         017920
C                               BODY M=ITOPOL(1,NOH) FIXED COORDINATES    017921
C                               IN BODY M DEFORMED STATE                  017922
C        ((RH(I,J,LRI+4),I=1,3),J=1,3) = TRANSFORMATION MATRIX Q-FRAME    017923
C                               COORDINATES TO P-FRAME COORDINATES AT     017924
C                               HINGE POINT NOH                           017925
C        ((RH(I,J,LRI+5),I=1,3),J=1,3) = TRANSFORMATION MATRIX FROM       017926
C                               BODY N=ITOPOL(1,NOH) FIXED COORDINATES    017927
C                               TO BODY M=ITOPOL(2,NOH) FIXED             017928
C                               COORDINATES IN THE DEFORMED STATE OF      017929
C                               BODIES N AND M                            017930
C     ROL     = BODY FIXED TO INERTIAL FRAME TRANSFORMATION MATRIX ARRAYS 017931
C               DIMENSIONED (3,3,NOMAX), STORED IN /3HBSRO/               017932
C         ((ROL(I,J,N),I=1,3),J=1,3) = 9 ELEMENTS OF THE TRANSFORMATION   017933
C                                       MATRIX USED TO TRANSFORM VECTORS  017934
C                                       FROM BODY N FIXED REFERENCE FRAME 017935
C                                       COORDINATES TO THE INERTIAL FIXED 017936
C                                       REFERENCE FRAME                   017937
C     DOL     = POSITION VECTOR ARRAY, INERTIAL ORIGIN TO BODY FIXED      017938
C               FRAME ORIGIN, DIMENSIONED (3,NOMAX), STORED IN /3HBSRO/   017939
C                  (DOL(I,N),I=1,3) = 3 COMPONENTS OF THE POSITION        017940
C                                       VECTOR FROM THE INERTIAL ORIGIN   017941
C                                       TO THE ORIGIN OF THE BODY N FIXED 017942
C                                       REFERENCE WRT THE INERTIALLY      017943
C                                       FIXED REFERENCE                   017944
C     RIN     = RH INVERSE TRANSFORMATIONS FOR HINGE POINT KINEMATICS     017945
```

```
C              DIMENSIONED (3,3,NHMAX), STORED IN /PINRP/.                 017946
C              SEE APPENDIX D VOL I                                        017947
C        ((PIN(I,J,NOH),I=1,3),J=1,3) = PI INVERSE TRANSFORMATION USED     017948
C                                TO RELATE THE EULER RATES ABOUT THE       017949
C                                SKEWED AXES AT HINGE POINT NOH TO THE     017950
C                                COMPONENTS OF THE RELATIVE ANGULAR        017951
C                                VELOCITY VECTOR OF THE Q-FRAME WRT THE    017952
C                                P-FRAME AT HINGE POINT NOH.               017953
C     HP2      = ARRAYS USED TO STORE MATRICES REQUIRED FOR THE            017954
C     HP3        COMPUTATION OF THE TIME DERIVATIVE OF THE PI INVERSE      017955
C                MATRIX, BOTH DIMENSIONED (3,3,NHMAX) AND STORED           017956
C                IN /PINRP/                                                017957
C        ((HP2(I,J,NOH),I=1,3),J=1,3) = 3 BY 3 COEFFICIENT MATRIX FOR      017958
C                                THETA SUB 2 DOT IN EQ C-16 VOL 1 USED     017959
C                                TO GET PI INVERSE DOT AT HINGE NOH        017960
C        ((HP3(I,J,NOH),I=1,3),J=1,3) = 3 BY 3 COEFFICIENT MATRIX FOR      017961
C                                THETA SUB 3 DOT IN EQ C-16 VOL 1 USED     017962
C                                TO GET PI INVERSE DOT AT HINGE NOH        017963
C                                                                          017964
C                                                                          017965
C        RETDS - SENSOR FRAME LOCATION AND ORIENTATION                     017966
C        -----------------------------------------                         017967
C     RS       = TRANSFORMATION MATRICES FROM SENSOR POINT TO BODY FIXED   017968
C                REFERENCE FRAME, DIMENSIONED (3,3,2*NSPMAX), STORED       017969
C                IN /SPECIF/. SET DEBUG(46) = .TRUE. TO PRINT              017970
C     DS       = COORDINATES OF POSITION VECTOR FROM BODY FIXED REFERENCE  017971
C                TO SENSOR POINT WRT TO BODY FRAME, STORED IN /SPECIF/     017972
C                DIMENSIONED (3,2*NSPMAX). SET DEBUG(46) = .TRUE. TO PRINT 017973
C                  FOR SENSOR POINT NS, NS=1,2,...,NSPT LET                017974
C                       N = IFTSMA(NS)                                     017975
C                     LR1 = 2*NS-1                                         017976
C                     LR2 = 2*NS                                           017977
C        ((RS(I,J,LR1),I=1,3),J=1,3) = TRANSFORMATION MATRIX WHICH         017978
C                            TAKES VECTORS FROM SENSOR POINT NS FIXED      017979
C                            COORDINATES TO BODY N FIXED COORDINATES       017980
C                            IN THE UNDEFORMED STATE OF BODY N             017981
C        ((RS(I,J,LR2),I=1,3),J=1,3) = TRANSFORMATION MATRIX SENSOR        017982
C                            POINT NS FRAME TO BODY N FRAME IN DEFORMED    017983
C                            STATE OF BODY N                               017984
C        (DS(I,LR1),I=1,3)             = POSITION VECTOR FROM BODY N       017985
C                            REFERENCE POINT TO SENSOR POINT NS WRT THE    017986
C                            BODY N FIXED REFERENCE IN UNDEFORMED STATE    017987
C        (DS(I,LR2),I=1,3)             = POSITION VECTOR FROM BODY N       017988
C                            REFERENCE POINT TO SENSOR POINT MS WRT THE    017989
C                            BODY N FIXED REFERENCE IN DEFORMED STATE      017990
C                                                                          017991
C                                                                          017992
C        EHGENR - HINGE POINT KINEMATICS COEFFICIENTS                      017993
C        ----------------------------------------                          017994
C                                                                          017995
C     HH       = VELOCITY TRANSFORMATION ARRAY. THE NON-ZERO PARTITIONS    017996
C                OF THE VELOCITY TRANSFORMATION EQUATION II-84 ARE STORED  017997
C                IN THIS ARRAY. THE ARRAY IS STORED IN /DSBSRD/ AND        017998
C                DIMENSIONED (6,6+NMDRCD,2*NHMAX-1).                       017999
C                SET DEBUG(45) = .TRUE. TO PRINT                           018000
C        ((HH(I,J,1),I=1,6),J=1,6+IRGFLX(1)) = COMPONENTS OF THE (1,1)     018001
C                            PARTITION OF THE COEFFICIENT MATRIX IN        018002
C                            EQUATION II-84. RELATES TO THE Q-FRAME IN     018003
C                            BODY 1 AT HINGE POINT 1, EVALUATED BY USE     018004
C                            OF EQUATION II-83                             018005
```

```
C               AT HINGE POINT NOH, NOH=2,3,.....NH C.T                      018006
C                            LQ = 6*NOH-2                                    018007
C                            LP = 6*NOH-1                                    018008
C                             N = ITOPOL(1,NOH)  (Q-FRAME IN BODY N)         018009
C                             M = ITOPOL(2,NOH)  (P-FRAME IN BODY M)         018010
C                           NMQ = IROPEX(N)      (NMQ MODES FOR BODY N)      018011
C                           NMP = IROPEX(M)      (NMP MODES FOR BODY M)      018012
C         ((HP(I,J,LP),I=1,6),J=1,6+NMP) = COMPONENTS OF THE (NOH,M)         018013
C                              PARTITION OF THE COEFFICIENT MATRIX IN        018014
C                              EQUATION II-94, RELATES TO THE P-FRAME IN     018015
C                              BODY M AT HINGE POINT NOH, EVALUATED BY       018016
C                              USE OF EQUATION II-82                         018017
C         ((HP(I,J,LQ),I=1,6),J=1,6+NMQ) = COMPONENTS OF THE (NOH,N)         018018
C                              PARTITION OF THE COEFFICIENT MATRIX IN        018019
C                              EQUATION II-92, RELATES TO THE Q-FRAME IN     018020
C                              BODY N AT HINGE POINT NOH, EVALUATED BY       018021
C                              USE OF EQUATION II-82                         018022
C                                                                            018023
C                                                                            018024
C         COMMON - SENSOR POINT KINEMATICAL COEFFICIENTS                     018025
C         ----------------------------------------------                     018026
C     ES        VELOCITY TRANSFORMATION MATRIX FOR SENSOR POINT FRAMES       018027
C              DIMENSIONED (6,6+MAXMOD,NSPT) STORED IN /DBBSRD/              018028
C               SET DEBUG(47) = .TRUE. TO PRINT                              018029
C                    FOR SENSOR POINT NS, NS=1,2,....NSPT                    018030
C                          N = IFISNA(NS)  SENSOR POINT NS IN BODY N         018031
C                        LE = IROPEX(N)   LE MODES FOR BODY N                018032
C         ((ES(I,J,NS),I=1,6),J=1,6+LE) = TRANSFORMATION MATRIX USED TO      018033
C                              OBTAIN THE COMPONENTS OF THE BODY N           018034
C                              INERTIAL ANGULAR AND LINEAR VELOCITY          018035
C                              VECTORS IN SENSOR NS FRAME COORDINATES        018036
C                              SEE PAGE 17, VOL II                           018037
C                                                                            018038
C     USER DEFINE CONTROL SYSTEM DATA                                        018039
C---------------------------------------                                     018040
C                                                                            018041
C     CNTDTA = REAL*8 ARRAY OF CONSTANTS READ IN, IN DYNSID AND STORED       018042
C              IN /CONPAR/. THESE ARE USER DEFINED PARAMETERS.               018043
C              DIMENSIONED (MAXCNT)                                          018044
C              DISCOS ASSUMES THAT:                                          018045
C                CNTDTA(I),I=1,NDELTA  = INITIAL CONDITIONS FOR THE          018046
C                                        NDELTA DIFFERENTIAL EQUATIONS       018047
C                                        CODED BY THE USER IN CONTRL         018048
C                CNTDTA(I),I.GT.NDELTA = CONSTANTS THAT THE USER WILL        018049
C                                        NEED FOR THE USER CODED             018050
C                                        EQUATIONS                           018051
C                                                                            018052
C                                                                            018053
C     STATE VECTOR AND ITS FIRST TIME DERIVATIVE                             018054
C-----------------------------------------------                             018055
C                                                                            018056
C     THE REAL*8 ARRAYS Y AND YDT ARE STORED IN /VECTOR/ AND                 018057
C        DIMENSIONED (KY). IN SEVERAL ROUTINES THE SYMBOLIC NAME YD          018058
C        IS USED INSTEAD OF YDT. SET DEBUG(39)=.TRUE. TO PRINT               018059
C        INSTANTANEOUS VALUES OF ELEMENTS IN Y AND YDT ARRAYS                018060
C                                                                            018061
C     (Y(I),I=LOCO(N),LOCO(N)+2)  = COMPONENTS OF INERTIAL ANGULAR           018062
C                                   VELOCITY VECTOR OF BODY N WRT THE        018063
C                                   BODY N REFERENCE                         018064
C     (Y(I),I=LOCO(N)+3,LOCO(N)+5)= COMPONENTS OF INERTIAL LINEAR            018065
```

```
C                                         VELOCITY VECTOR OF BODY N WRT THE          018066
C                                         BODY N REFERENCE                           018067
C     (Y(I),I=LOCU(N)+6,LOCU(N)+6 = GENERALIZED MODAL VELOCITIES               018068
C                -1+IRGFLX(N))            ASSOCITATED WITH THE FLEXURAL MODES        018069
C                                         OF BODY N. (IN ASCENDING ORDER AS          018070
C                                         READ IN, NONE IF BODY N RIGID)             018071
C     (Y(I),I=LOCU(N)+6+IRGFLX(N),= SPIN RATES OF THE ACTIVE WHEELS              018072
C                LOCU(N)+6+IRGFLX(N)      IN BODY N. (IN ASCENDING ORDER AS          018073
C                  -1+NMOM(2,N))          READ IN, NONE IF NMOM(2,N)=0)              018074
C     (Y(I),I=LOCU(NB+N),               = GENERALIZED MODAL DISPLACEMENTS            018075
C          LOCU(NB+N)-1+IRGFLX(N))        ASSOCIATED WITH THE FLEXURAL MODES         018076
C                                         OF BODY N. (IN ASCENDING ORDER AS          018077
C                                         READ IN, NONE IF BODY N RIGID)             018078
C     (Y(I),I=LOCU(2*NB+1),             = BETA COORDINATES USED TO DEFINE            018079
C            LOCU(2*NB+1)-1+NBETA)        UNCONSTRAINED AND RHENOMICALLY             018080
C                                         CONSTRAINED MOTION BETWEEN BODIES          018081
C                                         WHICH ARE CONTIGUOUS.                      018082
C                                          LET:                                      018083
C                                             NUM=BETA COORDINATE NUMBER FOR         018084
C                                                 DEGREE OF FREEDOM I-1 AT           018085
C                                                 HINGE POINT N. THE                 018086
C                                                 NUMBERING ALGORITHM IS             018087
C                                            NUM = 0                                 018088
C                                            DO 1 N=1,NH                             018089
C                                            DO 1 I=2,7                              018090
C                                            IF(IHDATA(I,N).EQ.1) GO TO 1            018091
C                                            NUM = NUM+1                             018092
C                                          1 CONTINUE                                018093
C     (Y(I),I=LOCU(2*NB+2),             = STATE VARIABLE COORDINATES WHICH           018094
C            LOCU(2*NB+2)-1+NDELTA)       ARE ASSOCIATED WITH THE NDELTA FIRST       018095
C                                         ORDER DIFFENTIAL EQUATIONS CODED BY        018096
C                                         THE USER IN CONTRL                         018097
C     (YD(I),I=1,NEQ)                   = FIRST TIME DERIVATIVE OF THE               018098
C                                         COORDINATE Y(I) IN THE STATE VECTOR        018099
C                                                                                    018100
C                                                                                    018101
C          FINDQ - FIND SET OF INDEPENDENT COORDINATES IN STATE VECTOR               018102
C          -----------------------------------------------------------               018103
C                                                                                    018104
C     IFLAG  = DUMMY VARIABLE SET IN CALLING ROUTINE                                 018105
C              IFLAG = 0 INITIAL PASS THROUGH COMPUTE MAGNITUDES OF                  018106
C                        INDEPENDENT COORDINATES OF STATE VECTOR                     018107
C              IFLAG = 1 NOT AN INTERMEDIATE STEP THROUGH RUNGE-KUTTA                018108
C                        INTEGRATION CYCLE. GET THE DEPENDENT COORDINATE             018109
C                        MAGNITUDES AND PUT THEM IN THE STATE VECTOR.                018110
C                        ALSO CHECK TO SEE IF OPTIMUM SET OF INDEPENDENT             018111
C                        COORDINATES BEING USED                                      018112
C              IFLAG = 2 AN INTERMEDIATE STEP THROUGH RUNGE-KUTTA,GET                018113
C                        DEPENDENT COORDINATES. USE SAME SET OF                      018114
C                        INDEPENDENT COORDINATES AS COMPUTED ON LAST                 018115
C                        PASS THROUGH WITH IFLAG=1                                   018116
C     UW     = MULTI-PURPOSE ARRAY STORED IN /AMDUM/                                 018117
C              DIMENSIONED (6*NHMAX,NBMAX*(6+NMODBD)+NWMAX).                         018118
C              AT THE DEBUG#30 PRINT POINT IN FINDQ LABELLED                         018119
C              'THE UW MATRIX IS'                                                    018120
C              ((UW(I,J),I=1,6*NH),J=1,NU) = THE VELOCITY TRANSFORMATION             018121
C                                      MATRIX AS DEFINED BY EQUATION II-84           018122
C                                      IN VOL I                                      018123
C     INDEP  = INDEPENDENT/DEPENDENT COORDINATE TABLE, AN INTEGER*4                  018124
C              ARRAY STORED IN /VINDEP/ AND DIMENSIONED (KY). USED FOR               018125
```

```
C               THE CASES WHEN                                                018126
C                               IFLAG = 1 OR 2                                018127
C               TO DEFINE WHICH COORDINATES IN THE STATE VECTOR Y ARE         018128
C               TO BE TREATED AS BEING INDEPENDENT AND WHICH DEPENDENT        018129
C                  INDEP(LU) = 0 IMPLIES THE COORDINATE Y(LU) IS FOR THE      018130
C                                PRESENT INTEGRATION STEP A DEPENDENT         018131
C                                COORDINATE                                   018132
C                  INDEP(LU) = 1 IMPLIES THE COORDINATE Y(LU) IS FOR THE      018133
C                                PRESENT INTEGRATION STEP AN INDEPENDENT      018134
C                                COORDINATE LU=1,...,NU                       018135
C               AT THE END OF EACH INTEGRATION STEP A CHECK IS MADE IN        018136
C               FINDU TO ASCERTIAN IF A BETTER SET OF INDEPENDENT             018137
C               COORDINATES CAN BE FOUND; IF SO, ARRAY INDEP IS UPDATED       018138
C          THE ALGORITHM FOR COMPUTING THE SET OF INDEPENDENT COORDS.         018139
C               HAS A BUILT IN BIAS. THE DEPENDENT COORDS. USED TO            018140
C               ACCOUNT FOR CONSTRAINTS AT HINGE POINT I ARE CHOOSEN          018141
C               FROM THE VELOCITY COMPONENTS USED TO DEFINE THE INERTIAL      018142
C               VELOCITY OF THE Q FRAME RELATIVE TO THE P FRAME AT            018143
C               HINGE POINT I. IF THIS CAN'T BE DONE ANY COORDINATE THEN      018144
C               BECOMES FAIR GAME FOR USE.                                    018145
C                                                                             018146
C                                                                             018147
C          YDOT - FIRST TIME DERIVATIVE OF STATE VECTOR                       018148
C          ----------------------------------------                           018149
C       SUBROUTINE YDOT DIRECTS THE COMPUTATION REQUIRED FOR THE              018150
C          DETERMINATION OF THE YDT ARRAY.                                    018151
C       GGV     = TORQUE ARRAY, DIMENSIONED (NBMAX*(6+NMOBOD)+NMWMAX)         018152
C               THIS ARRAY IS NOT STORED IN COMMON. IN YDOT IT IS             018153
C               INITIALIZED TO ZERO ON EACH PASS THROUGH. IN SUBROUTINE       018154
C               GRVGRD THE GENERALIZED FORCE ASSOCIATED WITH GRAVITY          018155
C               GRADIENT EFFECTS ACTING ON EACH STATE VECTOR                  018156
C               COORDINATE IS COMPUTED AND STORED IN GGV. IN SUBROUTINE       018157
C               TORQUE ALL OTHER SYSTEM FORCE AND TORQUES ARE COMPUTED        018158
C               AND ADDED ON TO GGV, DEBUG(50 AND 57) = .TRUE. TO PRINT       018159
C          (GGV(J),J=1,NU) = TOTAL GENERALIZED FORCE ASSOCIATED WITH THE      018160
C                             J-TH STATE VECTOR COORDINATE EQUATION           018161
C       BMB     = B*M**(-1)*B**T MATRIX IN EQUATION II-6, VOL 1, COMPUTED     018162
C               IN GETBMB AND PRINTED IF DEBUG(48)=.TRUE., STORED             018163
C               TEMPORARILY IN THE BW ARRAY VIA AN EQUIVALENCE                018164
C       BM      = B*M**(-1) MATRIX USED ALSO IN EQUATION II-6, COMPUTED       018165
C               IN GETBMB AND STORED TEMPORARILY IN ANOTHER PART OF           018166
C               THE BW ARRAY                                                  018167
C       BDTP    = FIRST DERIVATIVE OF THE HINGE POINT KINEMATIC COEFFICIENT   018168
C               MATRIX ASSOCIATED WITH THE P-FRAME, D/DT(EQ. C-1, VOL 1)      018169
C               DIMENSIONED (6,6+NMOBOD), STORED LOCALLY                      018170
C               BDTP FOR EACH HINGE POINT PRINTED IF DEBUG(51)=.TRUE.         018171
C       BDTQ    = FIRST DERIVATIVE OF THE HINGE POINT KINEMATIC COEFFICIENT   018172
C               MATRIX ASSOCIATED WITH THE Q-FRAME, D/DT(EQ. C-2, VOL 1)      018173
C               DIMENSIONED (6,6+NMOBOD), STORED LOCALLY                      018174
C               BDTQ FOR EACH HINGE POINT PRINTED IF DEBUG(51)=.TRUE.         018175
C       ALAM    = LAGRANGE MULTIPLIERS, FORCES AND TORQUES OF CONSTRAINT      018176
C               STORED IN A COMPACT ARRAY IN /LAMBDA/ WHICH IS                018177
C               DIMENSIONED (6*NHMAX), DEBUG(52) =.TRUE. TO PRINT             018178
C                    LET   IC = IV(I,L)  THEN                                 018179
C          ALAM(IC) = CONSTRAINT FORCE OR TORQUE ACTING TO RESTRICT           018180
C                     BETAH(I,L) MOTION AT HINGE POINT L. NOTE THAT           018181
C                     THE LAGRANGE MULTIPLIERS COMPUTED IN BAKSLV ARE         018182
C                     THE CONSTRAINT FORCES AND TORQUES                       018183
C       V       = ALAM VIA AN EQUIVALENCE IN YDOT                             018184
C       IV      = LAGRANGE MULTIPLIER NUMBERING TABLE, INTEGER ARRAY STORED   018185
```

```
C               IN /ZIVCNS/ DIMENSIONED (6,NHMAX) TO PROVIDE A COMPACT     018186
C                NUMBERING SEQUENCE FOR THE LAGRANGE MULTIPLIERS           018187
C        IVC(I,L) = NUMBER ASSIGNED INTERNALLY TO THE LAGRANGE             018188
C                   MULTIPLIER ASSOCIATED WITH DEGREE OF FREEDOM I         018189
C                   AT HINGE POINT L, EQUALS ZERO IF DEGREE OF             018190
C                   FREEDOM I AT HINGE POINT L UNCONSTRAINED.              018191
C                   COMPUTED IN SETIVC                                     018192
C                                                                          018193
C                                                                          018194
C     USER SUPPLIED SUBROUTINES                                            018195
C--------------------------                                                018196
C                                                                          018197
C     RATHER THAN ATTEMPT THE IMPOSSIBLE TASK OF FORSEEING ALL POSSIBLE    018198
C         SIMULATION NEEDS OF ALL POTENTIAL USERS                          018199
C                                                                          018200
C                   'DISCOS DEMANDS USER INTERFACE'                        018201
C                                                                          018202
C     THE USER MUST CODE IN FORTRAN ALL EQUATIONS REQUIRED TO DESCRIBE     018203
C         ALL NON-GYROSCOPIC AND NON-GRAVITY GRADIENT EFFECTS REQUIRED     018204
C         FOR SIMULATION PURPOSES                                          018205
C                                                                          018206
C     DISCOS ATTEMPTS TO PROVIDE THE USER WITH A CLEAN INTERFACE BETWEEN   018207
C         PARAMETERS WHICH WOULD NORMALLY BE CONSIDERED TO BE PHYSICALLY   018208
C         REALIZABLE AND THE GENERALIZED FORCE AND TORQUE PARAMETERS       018209
C         USED FOR ACTUAL COMPUTATION. THIS IS DONE VIA A SET OF           018210
C         FUNCTIONS AND SUBROUTINES WHICH REQUIRE THE USER TO DEFINE       018211
C         CERTIAN PHYSICALLY REALIZABLE PARAMETERS; THESE PARAMETERS       018212
C         ARE THEN AUTOMATICALLY TRANSFORMED INTO THE APPROPRIATE          018213
C         GENERALIZED PARAMETERS REQUIRED FOR COMPUTATION.                 018214
C                                                                          018215
C     ALL USER REQUIRED INPUT DATA IS NORMALLY READ FROM CARDS BY          018216
C         FORMATTED READ STATEMENTS IN SUBROUTINE DYNSIO. IT IS HERE       018217
C         THAT THE USER DEFINES                                            018218
C          1) NDELTA - TOTAL NUMBER OF ADDITIONAL FIRST ORDER              018219
C                      DIFFERENTIAL EQUATIONS WHICH WILL BE USER           018220
C                      DEFINED IN SUBROUTINE CONTRL                        018221
C          2) CNTDTA(I),I.LE.NDELTA - INITIAL CONDITIONS FOR THE USER      018222
C                                     DEFINED DIFFERENTIAL EQUATIONS       018223
C          3) CNTDTA(I),I.GT.NDELTA - ALL OTHER PARAMETERIC MAGNITUDES     018224
C                                     OF PARAMETERS USED IN THE USER       018225
C                                     DEFINED EQUATIONS                    018226
C                                                                          018227
C     ALL USER REQUIRED STATE VARIABLE DATA CAN BE OBTAINED BY SIMPLY      018228
C         INCLUDING THE APPROPRIATE COMMON BLOCKS IN THE SUBROUTINES       018229
C         BEING USER MODIFIED. DATA MAY BE TRANSFERED BETWEEN THE          018230
C         USER MODIFIABLE ROUTINES BY EITHER USING AVAILABLE SPACE         018231
C         IN /ZCONPAR/ OR BY DEFINING NEW SPECIAL PURPOSE COMMON BLOCKS    018232
C                                                                          018233
C                                                                          018234
C         SUBROUTINE CONTRL   USER DEFINED DIFFERENTIAL EQUATIONS          018235
C         -------------------------------------------------------          018236
C                                                                          018237
C     USER SPECIFICATION OF ALL NON-GYROSCOPIC AND NON-GRAVITY GRADIENT    018238
C         FORCES AND TORQUES FREQUENTLY REQUIRES THE CAPABILITY TO ADD     018239
C         AND SIMULTANEOUSLY SOLVE SEVERAL ADDITIONAL DIFFERENTIAL         018240
C         EQUATIONS. THESE ARE REFERED TO AS 'CONTROL EQUATIONS' SINCE     018241
C         IN MOST SPACECRAFT SIMULATION PROBLEMS THEY ARE THE EQUATIONS    018242
C         WHICH SIMULATE THE ACTIVE ON-BOARD CONTROL SYSTEM. THE TOTAL     018243
C         NUMBER OF USER DEFINED FIRST ORDER DIFFERENTIAL EQUATIONS        018244
C         IS 'NDELTA'. THIS PARAMETER IS SET VIA DATA INPUT IN DYNSIO      018245
```

```
C      *** IF TRANSFER FUNCTIONS ARE TO BE GENERATED THE USER MUST ADHERE  018246
C          TO THE RULE THAT ALL CONTROL INFORMATION (SYSTEM STATE DATA)    018247
C          COMES THROUGH SENSOR SIGNALS. IF THIS IS NOT DONE THE           018248
C          NUMERICAL LINEARIZATION ALGORTHIM CANNOT DIFFERENTIATE          018249
C          BETWEEN CONTROL TORQUES AND GYROSCOPIC TORQUES, AND HENCE       018250
C          CANNOT OPEN FEEDBACK LOOPS WHEN OPEN AND QUASI-OPEN LOOP        018251
C          TRANSFER FUNCTIONS ARE CALLED FOR                               018252
C                                                                          018253
C      IN ADDITION TO DEFINING ALL CONTROL EQUATIONS IN CONTRL INTERNAL    018254
C          COMPUTATION LOGIC REQUIRES THAT THE USER ALSO DEFINE IN CONTRL  018255
C                NXSS - TOTAL NUMBER OF SENSOR SIGNALS TO BE DEFINED       018256
C                       IN EQADD (DEFAULT NXSS=1)                          018257
C                NBTQ - TOTAL NUMBER OF TORQUE SIGNALS TO BE DEFINED       018258
C                       IN EQADD (DEFAULT NBTQ=0)                          018259
C                                                                          018260
C      DISCOS ASSUMES THAT THE NDELTA STATE VARIABLES ASSOCIATED WITH THE  018261
C          NDELTA USER DEFINED CONTROL EQUATIONS ARE INDEPENDENT. TO       018262
C          SIMULTANEOUSLY SOLVE THESE EQUATIONS ALONG WITH THE DISCOS      018263
C          DERIVED PLANT EQUATIONS THE FOLLOWING STORAGE CONVENTION MUST   018264
C          BE ADHERED TO:                                                  018265
C           LET                                                            018266
C                LOCL = LOCO(2*NB+2)-1  (SEE DEFINITION ABOVE OF LOCO)     018267
C            THEN                                                          018268
C              YDT(LOCL+I),I=1,NDELTA = USER DEFINED LINEAR OR NON-LINEAR  018269
C                                       FUNCTION OF TIME AND SYSTEM STATE. 018270
C                                       THIS IS THE I-TH USER DEFINED      018271
C                                       FIRST ORDER DIFFERENTIAL EQUATION  018272
C                                       WHICH WILL BE SIMULATEOUSLY        018273
C                                       INTEGRATED WITH THE DISCOS DERIVED 018274
C                                       PLANT EQUATIONS                    018275
C                Y(LOCL+I),I=1,NDELTA = LOCATION IN THE STATE VECTOR ARRAY 018276
C                                       OF THE STATE VARIABLE OBTAINED VIA 018277
C                                       NUMERICAL INTEGRATION OF THE I-TH  018278
C                                       USER DEFINED DIFFERENTIAL EQUATION 018279
C           AND                                                            018280
C               CNTDTA(I),I=1,NDELTA = INITIAL CONDITION TO BE USED FOR    018281
C                                       THE I-TH USER DEFINED DIFFERENTIAL 018282
C                                       EQUATION. THESE PARAMETERS ARE SET 018283
C                                       VIA DATA INPUT IN DYNSIO           018284
C                                                                          018285
C      FREQUENTLY THE ONLY CONTROL SYSTEM DESCRIPTION AVAILABLE TO THE     018286
C          ANALYSIST IS A MULTILOOP FEEDBACK BLOCK DIAGRAM CONSISTING      018287
C          OF TRANSFER FUNCTIONS, LOGIC GATES AND NON-LINEAR ELEMENTS.     018288
C          THE DISCOS USER MAY EITHER TRANSFORM THIS DISCRIPTION TO THE    018289
C          TIME DOMAIN AND DIRECTLY CODE THE RESULTANT DIFFERENTIAL        018290
C          EQUATIONS, OR SIMPLY INPUT THE VARIOUS TRANSFER FUNCTIONS AND   018291
C          LET THE COMPUTER TRANSFORM THEM AND SET UP THE APPROPRIATE      018292
C          DIFFERENTIAL EQUATIONS. THE USER MAY IF IT IS DESIRABLE CODE    018293
C          DIRECTLY SOME OF THE DIFFERENTIAL EQUATIONS AND LET THE         018294
C          COMPUTER GENERATE THE REST FROM INPUTTED TRANSFER FUNCTION      018295
C          DATA. OBVIOUSLY, THE USER MUST ADHERE TO CERTIAN PROGRAMMING    018296
C          RULES TO SET THIS CAPABILITY UP:                                018297
C          1) THERE IS NO REAL LIMIT AS TO THE NUMBER OF TRANSFER          018298
C             FUNCTIONS TO BE INPUTTED OR TO THE DEGREE OF THEIR           018299
C             DENOMINATOR POLYNOMIALS. THIS IS USER SET IN CONTRL. EACH    018300
C             TRANSFER FUNCTION MUST BE GIVEN AS A POLYNOMIAL RATIO IN S   018301
C             OF THE FORM                                                  018302
C                                                                          018303
C                       A(1) + A(2)*S + ... + A(KS)*S**(KS-1)              018304
C                TF(S) = ------------------------------------              018305
```

```
C                       B(1) + B(2)*S + ... + B(NS)*S**(NS-1)
C
C           WHERE            (KS.LE.NS.AND.NS.GE.2)
C        2) DEFINE THE FOLLOWING MAXIMUM SIZE PARAMETERS VIA DATA
C           STATEMENTS IN SUBROUTINE CONTRL
C             NPLY = TOTAL NUMBER OF TRANSFER FUNCTIONS TO BE INPUTTED
C             IIST = 0, FIRST PASS THROUGH CONTRL FLAG
C             KRY  = 1+THE DEGREE OF THE HIGHEST DEGREE DENOMINATOR
C                    POLYNOMIAL TO BE INPUTTED (USED TO SET DIMENSION
C                    OF ARRAY IN WHICH ALL POLY. COEFF. STORED)
C             KCY  = 2*NPLY
C        3) USE THE FOLLOWING ACRONYMS TO DEFINE THE TRANSFER FUNCTION
C           DATA STORAGE ARRAYS
C             KPLY = TRANSFER FUNCTION DENOMINATOR DEGREE ARRAY,
C                    INPUTTED INTEGER*4 ARRAY DIMENSIONED (NPLY)
C                      KPLY(K) = ONE PLUS THE DEGREE OF THE DENOMINATOR
C                                OF THE K-TH INPUTTED TRANSFER
C                                FUNCTION, K=1,2,...,NPLY
C             CPLY = TRANSFER FUNCTION POLYNOMIALS COEFFICIENTS ARRAY,
C                    INPUTTED REAL*8 ARRAY DIMENSIONED (KRY,KCY)
C                      CPLY(I,2*K-1) = COEFFICIENT OF THE S**(I-1) TERM
C                                      IN THE DENOMINATOR OF THE K-TH
C                                      INPUTTED TRANSFER FUNCTION
C                      CPLY(I,2*K)   = COEFFICIENT OF THE S**(I-1) TERM
C                                      IN THE NUMERATOR OF THE K-TH
C                                      INPUTTED TRANSFER FUNCTION
C                                          K=1,2,...,NPLY
C                                          I=1,2,...,KPLY(K)
C        4) INSERT THE FOLLOWING FORTRAN CODE IN CONTRL TO INPUT ALL
C           TRANSFER FUNCTION DATA IN A DISCOS CONSISTANT FORMAT
C                  IF(IIST.NE.0) GO TO 110
C                  IIST = 1
C                  IF(NPLY.EQ.0) GO TO 110
C                  CALL ZERO(CPLY,KRY,KCY,KRY)
C                  DO 105 K=1,NPLY
C                  K2 = 2*K-1
C              105 CALL READ(CPLY(I,K2),KPLY(K),K2,KRY,KCY)
C                  CALL WRITE(CPLY,KRY,KCY,4HCPLY,KRY)
C              110 CONTINUE
C        5) TO TRANSFORM S-DOMAIN TRANSFER FUNCTION DATA TO TIME DOMAIN
C           CONTROL EQUATIONS USE
C
C                       SUBROUTINE TFPLY(A,B,U,K,NS,X)
C
C           THE FOLLOWING BLOCK DIAGRAM SETUP IS ASSUMED BY TFPLY
C
C                         ---------
C               U(K,T)   |          |  X(K,T)
C              --------->| TF(K,S)  |---------->
C                        |          |
C                         ---------
C
C           TFPLY MUST BE CALLED EXACTLY NPLY TIMES FROM CONTRL EVERY
C           INTEGRATION STEP. ON THE K-TH CALL TO TFPLY FROM CONTRL
C           TFPLY ASSUMES THAT THE FOLLOWING PARAMETERS HAVE BEEN PUT
C           BY THE USER PROVIDED CODE INTO ITS DUMMY ARGUMENTS:
C             A = COEFFICIENTS OF THE DENOMINATOR POLYNOMIAL OF
C                  TF(K,S); THAT IS COLUMN 2*K-1 OF ARRAY CPLY
C             B = COEFFICIENTS OF THE NUMERATOR POLYNOMIAL OF
C                  TF(K,S); THAT IS COLUMN 2*K OF ARRAY CPLY
```

```
C            U   = INSTANTANEOUS VALUE OF THE USER DEFINED VARIABLE
C                  WHICH FEEDS INTO THE BLOCK DEFINED BY TRANSFER
C                  FUNCTION TF(K,S). U(K,T) MAY BE ANY LINEAR OR NON-
C                  LINEAR FUNCTION OF SYSTEM STATE VARIABLES DEFINED
C                  IN THE TIME DOMAIN
C            NS  = 1+DEGREE OF THE DENOMINATOR POLYNOMIAL OF TF(K,S);
C                  THAT IS, KPLY(K)
C            L   = LOCATION IN THE STATE VECTOR ARRAY Y OF THE LEADING
C                  ELEMENT OF THE KPLY(K)-1 STATE VARIABLES TO BE
C                  ESTABLISHED BY TFPLY FROM TF(K,S)
C                    NOTE THAT THE NPLY CALLS TO TFPLY WILL
C                    ESTABLISH EXACTLY
C                     KPLY(1)+KPLY(2)+...+KPLY(NPLY)-NPLY
C                    NEW CONTROL EQUATIONS AND ASSOCIATED
C                    STATE VARIABLES
C         WITH THE ABOVE DATA LOADED INTO THE DUMMY ARGUMENTS OF
C         TFPLY IT WILL DEFINE:
C           YD(L-1+I),I=1,KPLY(K)-1 = INSTANTANEOUS VALUE OF THE
C                                     FIRST TIME DERIVATIVE OF THE
C                                     I-TH STATE VARIABLE REQUIRED
C                                     BY THE TRANSFORMATION OF
C                                     S-DOMAIN DATA TO THE TIME
C                                     DOMAIN
C            X   = INSTANTANEOUS VALUE OF THE OUTPUT X(K,T) OF THE
C                  BLOCK DEFINED BY TRANSFER FUNCTION TF(K,S)
C         THE ALGORITHM USED TO GENERATE THE KPLY(K)-1 CONTROL
C        EQUATIONS IS GIVEN AS FOLLOWS: (SEE PAGE 78, VOL I)
C                  AN=A(NS)
C                  DO 10 I=1,NS
C                  A(I)=A(I)/AN
C               10 B(I)=B(I)/AN
C                  BN=B(NS)
C                  DO 20 I=2,NS
C                  J = NS-I+1
C                  K = L+I-2
C                  IF(I.EQ.NS) GO TO 25
C               20 YD(K) = -A(J)*Y(L) + Y(K+1) + (B(J)-A(J)*BN)*U
C               25 YD(K) = -A(J)*Y(L) + (B(J)-A(J)*BN)*U
C                  X = Y(L) + BN*U
C         THE USER WILL HAVE TO GO THROUGH THIS ALGORITHM IF NON-ZERO
C         INITIAL CONDITIONS FOR SELECTED CONTROL VARIABLES ARE
C         REQUIRED TO AVOID PROLONGED TRANSIENT RESPONSE. AGAIN, ALL
C         INITIAL CONDITIONS ARE DEFINED IN THE INPUT ARRAY CNTDTA
C
C
C      FUNCTIONS ADT & ADDT  DEFINE RHEONOMIC CONSTRAINTS
C      ---------------------------------------------------
C
C    ASSUME THAT AT HINGE POINT L, RELATIVE MOTION WITH RESPECT TO
C      DEGREE OF FREEDOM I IS RHEONOMICALLY CONSTRAINED.
C      LET
C                    NN = 6*(L-1) + I
C      THEN
C             ADDT(NN,T) = USER PRESCRIBED MAGNITUDE OF THE SECOND
C                          TIME DERIVATIVE OF BETAH(I,L) AT TIME T
C              ADT(NN,T) = USER PRESCRIBED MAGNITUDE OF THE FIRST
C                          TIME DERIVATIVE OF BETAH(I,L) AT TIME T
C      WHERE BETAH IS DEFINED ABOVE UNDER THE TITLE 'KINEMATIC DATA
C      INPUTTED AND COMPUTED'
C
```

```
C     IF ADDT(NN,I) IS TO BE DEFINED BY A DISCONTINUOUS FUNCTION OF TIME  018426
C         IT IS RECOMMENDED THAT ADT(NN,I) BE OBTAINED VIA NUMERICAL      018427
C         INTEGRATION OF ADDT(NN,I) IN SUBROUTINE CONTRL. IF THIS         018428
C         IS NOT DONE NUMERICAL PROBLEMS CAN OCCUR.                       018429
C                                                                         018430
C                                                                         018431
C         SUBROUTINE EXTOR   EXTERNAL TORQUES AND FORCES                  018432
C         -----------------------------------------------                 018433
C                                                                         018434
C     ALL EXTERNAL TORQUES AND FORCES ACTING ON THE SYSTEM MUST BE        018435
C         DEFINED IN SUBROUTINE EXTOR, SEE PAGE 12, VOL II. EXTOR IS      018436
C         CALLED FROM SUBROUTINE TORQUE IMMEDIATELY AFTER RETURN FROM     018437
C         CONTRL. UPON ENTRY INTO EXTOR THE STATE VECTOR Y CONTAINS THE   018438
C         INSTANTANEOUS VALUES OF ALL PLANT AND CONTROL SYSTEM STATE      018439
C         VARIABLES. THE USER MAY USE ANY OF THESE STATE VARIABLES OR     018440
C         ANY OTHER SYSTEM PARAMETERS STORED IN COMMON TO CREATE THE      018441
C         EXTERNAL FORCE/TORQUE VECTORS REQUIRED BY EQUATION II-74 OF     018442
C         VOL I.                                                          018443
C                                                                         018444
C     THE USER MUST ADHERE TO THE FOLLOWING RULES INORDER TO CORRECTLY    018445
C         DEFINE THE ACTION OF EXTERNAL FORCES AND TORQUES:               018446
C          1) EXTERNAL FORCES AND TORQUES MAY BE APPLIED ONLY AT          018447
C             'SENSOR POINTS', A MAXIMUM OF NSPMAX SUCH POINTS MAY BE     018448
C             DEFINED (VIA DATA INPUT IN DYNSID)                          018449
C          2) DEFINE THE FOLLOWING ACRONYMS AS                            018450
C              NTEX = TOTAL NUMBER OF 6-ELEMENT FORCE/TORQUE VECTORS      018451
C                     TO BE DEFINED IN EXTOR AND RETURNED TO TORQUE.      018452
C              ISPN = FORCE/TORQUE APPLICATION POINT ARRAY. THE FORCE/    018453
C                     TORQUE VECTORS MUST BE NUMBERED AND CORRELATED      018454
C                     WITH SENSOR POINTS. HENCE THE USER MUST DEFINE      018455
C                     THE ELEMENTS OF ISPN ON THE FIRST PASS THROUGH      018456
C                     EXTOR SUCH THAT                                     018457
C                          ISPN(L) = SENSOR POINT NUMBER ASSOCIATED WITH  018458
C                                    THE POINT AT WHICH FORCE/TORQUE      018459
C                                    VECTOR NUMBER L,L=1,2,....,NTEX      018460
C                                    IS TO BE APPLIED                     018461
C              TEX  = FORCE/TORQUE VECTOR ARRAY. FOR COMPUTATIONAL        018462
C                     EFFICIENCY THE USER SHOULD RETURN A MAXIMUM OF ONE  018463
C                     FORCE/TORQUE VECTOR PER SENSOR POINT, HENCE IT IS   018464
C                     DIMENSIONED (6,NSPMAX) IN TORQUE. THE ELEMENTS OF   018465
C                     FORCE/TORQUE VECTOR L ARE TO BE STORED IN COLUMN L  018466
C                     OF TEX ACCORDING TO THE FOLLOWING DEFINITIONS       018467
C                      TEX(I,L),I=1,2,3 = X,Y,Z COMPONENTS OF THE         018468
C                                         RESULTANT EXTERNAL TORQUE L TO  018469
C                                         BE APPLIED AT SENSOR POINT      018470
C                                         ISPN(L), RELATIVE TO THE LOCAL  018471
C                                         ISPN(L) FIXED REFERENCE FRAME   018472
C                      TEX(I,L),I=4,5,6 = X,Y,Z COMPONENTS OF THE         018473
C                                         RESULTANT EXTERNAL FORCE L TO   018474
C                                         BE APPLIED AT SENSOR POINT      018475
C                                         ISPN(L), RELATIVE TO THE LOCAL  018476
C                                         ISPN(L) FIXED REFERENCE FRAME   018477
C                     BOTH DISTRIBUTED AND LOCALLY APPLIED EXTERNAL       018478
C                     EFFECTS MAY BE CONSIDERED. THEY MAY BE ANY LINEAR   018479
C                     OR NON-LINEAR FUNCTION OF PLANT AND CONTROL SYSTEM  018480
C                     PARAMETERS; HOWEVER, ONLY RESULTANT FORCE/TORQUE    018481
C                     VECTORS ACTING AT PARTICULAR SENSOR POINTS ARE TO   018482
C                     BE RETURNED IN THE ARRAY TEX                        018483
C          3) UPON RETURNING TO TORQUE FROM EXTOR DISCOS MAKES USE OF     018484
C             THE DATA NOW CONTAINED IN THE ABOVE ARRAYS TO EVALUATE      018485
```

```
C              EQUATION II-74 OF VOL I WITH NO ADDITIONAL USER INTERFACE   018486
C                                                                          018487
C                                                                          018488
C         SUBROUTINE KHINGE  SPRINGS. DAMPERS AND MOTORS AT HINGES         018489
C         ---------------------------------------------------------        018490
C                                                                          018491
C     RELATIVE MOTION OF CONTIGUOUS BODIES IS OFTEN RESTRAINED             018492
C         BY MECHANISMS PLACED AT THEIR COMMON HINGE POINT. THESE          018493
C         MECHANISMS. WHICH MAY INCLUDE LINEAR OR NON-LINEAR SPRINGS.      018494
C         DAMPERS AND MOTORS. PRODUCE INTERNAL FORCES AND TORQUES WHICH    018495
C         ACT IN EQUAL AND OPPOSITE PAIRS ON THE CONTIGUOUS BODIES. ALL    018496
C         SUCH EFFECTS MUST BE USER DEFINED IN SUBROUTINE KHINGE. (SEE     018497
C         PAGE III-6 OF VOL II).                                           018498
C                                                                          018499
C     KHINGE PROVIDES THE USER WITH A CLEAN INTERFACE BETWEEN PARAMETERS   018500
C         WHICH ARE PHYSICALLY REALIZABLE AND PARAMETERS REQUIRED FOR      018501
C         COMPUTATION.                                                     018502
C                                                                          018503
C              RECALL THAT AT HINGE POINT L.L=1.2.....NH                   018504
C              EULER ANGLE ROTATION SEQUENCE TYPE IHDATA(1.L)              018505
C              IS USED TO ORIENT THE P-FRAME IN BODY ITOPOL(2.L)           018506
C              RELATIVE TO THE Q-FRAME IN BODY ITOPOL(1.L);                018507
C              FURTHERMORE DEGREE OF FREEDOM I.I=1.2....6 IS               018508
C              FREE OF EITHER FIXED OR RHEONOMIC CONSTRAINTS IF            018509
C              IHDATA(I+1.L)=0.                                            018510
C                                                                          018511
C     TO PROVIDE THE USER WITH THE DESIREABLE CLEAN INTERFACE A            018512
C         A MODELLING RESTRICTION MUST BE INTRODUCED; THAT IS.             018513
C                                                                          018514
C                INTERNAL TORQUE GENERATORS ARE ASSUMED TO                 018515
C                   ACT ABOUT EULER ANGLE ROTATION AXES                    018516
C                      WHICH ARE FREE OF FIXED OR                          018517
C                         RHEONOMIC CONSTRAINTS                            018518
C                                                                          018519
C         THE USER MUST PROVIDE DISCOS WITH THE COMPONENTS OF THE          018520
C         RESULTANT INTERNALLY GENERATED TORQUE RELATIVE TO THE SKEWED     018521
C         AXIS SYSTEM WHICH WAS USER SPECIFIED IN ARRAY IHDATA.            018522
C         FURTHERMORE.                                                     018523
C                                                                          018524
C                INTERNAL FORCE GENERATORS ARE ASSUMED TO                  018525
C                   ACT ALONG COORDINATE AXES OF THE                       018526
C                      Q-FRAME WHICH ARE FREE OF                           018527
C                         FIXED OR RHEONOMIC                               018528
C                            CONSTRAINTS                                   018529
C                                                                          018530
C         THE USER MUST PROVIDE DISCOS WITH THE COMPONENTS OF THE          018531
C         RESULTANT INTERNALLY GENERATED FORCE RELATIVE TO THE Q-FRAME     018532
C         COORDINATE AXES WHICH WAS USER SPECIFIED VIA INPUT DATA          018533
C                                                                          018534
C     THE USER MUST USE THE ACRONYM HNGT TO DEFINE THE ARRAY IN WHICH      018535
C         THE TORQUE AND FORCE COMPONENTS ARE STORED AND ADHERE TO         018536
C         THE FOLLOWING DEFINITIONS:                                       018537
C            AT HINGE POINT L.L=1.2.....NH                                 018538
C         HNGT(1.L) = INTERNAL TORQUE TO BE APPLIED ABOUT FIRST EULER      018539
C                     ROTATION AXIS AT HINGE POINT L                       018540
C         HNGT(2.L) = INTERNAL TORQUE TO BE APPLIED ABOUT SECOND EULER     018541
C                     ROTATION AXIS AT HINGE POINT L                       018542
C         HNGT(3.L) = INTERNAL TORQUE TO BE APPLIED ABOUT THIRD EULER      018543
C                     ROTATION AXIS AT HINGE POINT L                       018544
C         HNGT(4.L) = INTERNAL FORCE TO BE APPLIED ALONG THE X-AXIS OF     018545
```

```
C                    THE G-FRAME AT HINGE POINT L                        018546
C            FNGT(5,L) = INTERNAL FORCE TO BE APPLIED ALONG THE Y-AXIS OF  018547
C                    THE G-FRAME AT HINGE POINT L                        018548
C            FNGT(6,L) = INTERNAL FORCE TO BE APPLIED ALONG THE Z-AXIS OF  018549
C                    THE G-FRAME AT HINGE POINT L                        018550
C              FOR EXAMPLE ASSUME MOTION RELATIVE TO DEGREE OF FREEDOM   018551
C                      I,I=1,2,....,6 AT HINGE POINT L,L=1,2,....,NH      018552
C                      IS UNCONSTRAINED (IHDATA(I+1,L)=0) BUT THAT       018553
C                      IT IS RESTRAINED BY A LINEAR SPRING OF            018554
C                      STIFFNESS SK(I,L) AND A LINEAR DAMPER HAVING      018555
C                      A DAMPING CONSTANT DK(I,K). TO INCLUDE THIS       018556
C                      IN DISCOS THE USER SIMPLE CODES                   018557
C                HNGT(I,L) = -(SK(I,L)*BETAH(I,L) + DK(I,L)*BETAHD(I,L))  018558
C                      IN KHINGE. MOTOR GENERATED TORQUES (I=1,2,3)      018559
C                      OR FORCES (I=4,5,6) ARE SIMILARLY HANDLED. IN     018560
C                      MOST CASES THESE WOULD BE COMPUTED IN CONTRL      018561
C                      AND PASSED TO KHINGE IN A SPECIAL COMMON          018562
C                      BLOCK. IF TQFR(I,L) IS THE MOTOR TORQUE OR        018563
C                      FORCE RESTRAINING DEGREE OF FREEDOM I MOTION      018564
C                      AT HINGE POINT L ITS EFFECT WOULD BE INCLUDED     018565
C                      BY THE USER WRITTEN CODE                          018566
C                HNGT(I,L) = HNGT(I,L) + TQFR(I,L)                       018567
C                                                                        018568
C         ALL DATA REQUIRED BY THE USER IN KHINGE; SUCH AS, SPRING AND   018569
C             DAMPING CONSTANTS, ARE MOST CONVENIENTLY INPUTTED IN ARRAY 018570
C             CNTDTA. TIME DEPENDENT DATA REQUIRED BY THE USER; SUCH AS, 018571
C             MOTOR TORQUES OR FORCES, ARE MOST CONVENIENTLY COMPUTED IN 018572
C             KHINGE OR COMPUTED IN CONTRL AND PASSED TO KHINGE BY A SPECIAL 018573
C             PURPOSE COMMON BLOCK                                       018574
C                                                                        018575
C         IF MECHANISMS EXIST IN THE SYSTEM WHICH CAN STORE POTENTIAL ENERGY 018576
C             THEIR CONTRIBUTION TO TOTAL SYSTEM POTENTIAL ENERGY (TOTPE) 018577
C             MUST BE ACCOUNTED FOR. FOR THE LINEAR SPRING ASSUMED ABOVE  018578
C             ITS CONTRIBUTION IS INCLUDED BY THE USER WRITTEN CODE      018579
C                      TOTPE = TOTPE + .500*SK(I,L)*BETAH(I,L)**2        018580
C             CONTRIBUTIONS DUE TO NON-LINEAR SPRINGS MUST BE USER       018581
C             DETERMINED AND ALSO CODED. TOTAL SYSTEM POTENTIAL ENERGY IS 018582
C             NOT USED FOR ANY REQUIRED COMPUTATION IN DISCOS            018583
C                                                                        018584
C         UPON ENTRY INTO KHINGE CODE IS PROVIDED TO ZERO TOTPE AND ALL  018585
C             ELEMENTS OF THE ARRAY HNGT. THE USER THEN PROVIDES THE CODE 018586
C             TO DEFINE THE NON-ZERO ELEMENTS OF HNGT AND THE ELEMENT TOTPE. 018587
C             DISCOS PROCEEDS THEN VIA THE PROVIDED CODED ALGORITHM TO   018588
C             COMPUTE PARAMETERS REQUIRED FOR CONTINUING COMPUTATION.    018589
C                                                                        018590
C                                                                        018591
C             SUBROUTINE SHAFT   WHEEL TORQUES                           018592
C             ---------------------------------                          018593
C                                                                        018594
C         VARIABLE AND CONSTANT SPEED WHEELS MAY BOTH BE INCLUDED IN THE 018595
C             SIMULATION MODEL. WHEEL TORQUES MAY BE APPLIED ONLY TO THE 018596
C             VARIABLE SPEED WHEELS OF THE SYSTEM. THIS IS DONE IN A MANNER 018597
C             ALMOST IDENTICAL TO THAT USED IN SUBROUTINE KHINGE.        018598
C                                                                        018599
C         ONLY INTERNALLY GENERATED TORQUES MAY BE APPLIED TO WHEELS, AND 018600
C             THESE ACT ONLY ABOUT THE RESPECTIVE WHEEL AXES.            018601
C                                                                        018602
C               WHEEL I,I=1,......,NCFWG IS A VARIABLE SPEED WHEEL       018603
C                         IF(IMO(3,I).EQ.1)                              018604
C               WHERE THE IMO DATA ARRAY IS USER SPECIFIED VIA           018605
```

```
C           INPUT DATA IN DYNSIO                                      018606
C                                                                     018607
C     IN SUBROUTINE SHAFFT THE USER MUST USE THE ACRONYM TSHFT TO DEFINE 018608
C         THE ARRAY IN WHICH ALL WHEEL TORQUES ARE STORED AND ADHERE TO  018609
C         THE FOLLOWING DEFINITION FOR ELEMENT I OF THE ARRAY TSHFT      018610
C         WHICH IS DIMENSIONED (NWMAX) IN SUBROUTINE TORQUE              018611
C                                                                     018612
C           TSHFT(I) = INTERNALLY GENERATED TORQUE TO BE APPLIED ABOUT   018613
C                      THE SPIN AXIS OF WHEEL I. AS IN RHINGE THIS       018614
C                      TORQUE IS USUALLY COMPUTED IN CONTRL AND PASSED   018615
C                      TO SHAFFT VIA A SPECIAL PURPOSE COMMON BLOCK      018616
C                                                                     018617
C     THE COMPUTATION OF WHEEL TORQUE OFTEN REQUIRES KNOWLEDGE OF THE    018618
C         INSTANTANEOUS SPIN RATE AND AT TIMES, FOR SPECIAL SIMULATION   018619
C         PROBLEMS, WHEEL ANGLE. ALL SPIN RATES OF VARIABLE SPEED        018620
C         (ACTIVE) WHEELS ARE CONTAINED IN THE STATE VECTOR ARRAY Y. SEE 018621
C         DEFINITION ABOVE FOR EXACT LOCATION. SINCE WHEEL RATES ARE     018622
C         OFTEN VERY HIGH AND WHEEL ANGLE IS USUALLY NOT REQUIRED FOR    018623
C         SIMULATION PURPOSES WHEEL ANGLE IS NOT COMPUTED. IF REQUIRED,  018624
C         HOWEVER THE USER SIMPLE HAS TO ADD THE DIFFERENTIAL EQUATION   018625
C         IN CONTRL WHICH WILL COMPUTE IT.                               018626
C                                                                     018627
C     FOR EXAMPLE: IF THE SPIN OF WHEEL I IS DAMPED BY A LINEAR DAMPER   018628
C                  WITH DAMPING CONSTANT WDK(I) AND CONTROLLED BY A      018629
C                  MOTOR WHICH PRODUCES WHEEL TORQUE CLM(I) THE          018630
C                  USER MUST CODE                                        018631
C              IW = LOCATION IN ARRAY Y OF WHEEL I SPIN RATE. (A SIMPLE  018632
C                  INTEGER EQUATION BASED UPON ABOVE DEFINITION)         018633
C       TSHFT(I) = -WDK(I)*Y(IW) + CLM(I)                                018634
C                                                                     018635
C                                                                     018636
C         SUBROUTINE GMISC   MISCELLANEOUS AND THERMAL EFFECTS           018637
C         ---------------------------------------------------            018638
C                                                                     018639
C     THERMALLY INDUCED MOTION OF A FLEXIBLE BODY CAN OCCUR IF THERMAL   018640
C         TIME CONSTANTS AND FUNDAMENTAL NATURAL FREQUENCIES ARE OF      018641
C         NEARLY THE SAME ORDER OF MAGNITUDE. THE MOTION CAN AT TIMES    018642
C         BE SELF-SUSTAINING AND EVEN UNSTABLE IF THE DIRECTION AND      018643
C         MAGNITUDE OF THE INCIDENT THERMAL FIELD IS TIME VARYING        018644
C         RELATIVE TO A BODY FIXED REFERENCE. IN GENERAL THE FLEXIBLE    018645
C         BODY WILL MOVE TOWARD A TIME VARYING POSITION OF THERMAL       018646
C         EQUILIBRIUM WHICH WILL BE DEPENDENT UPON THE ORIENTATION OF    018647
C         THE BODY RELATIVE TO THE THERMAL FIELD                         018648
C                                                                     018649
C     THIS EFFECT CAN BE INCORPORATED IN SUBROUTINE GMISC. IN GMISC      018650
C         THE USER MUST HAVE AT HAND THE TIME VARYING MODAL AMPLITUDES   018651
C         OF THE THERMAL EQUILIBRIUM POSITION, USUALLY OBTAINED VIA      018652
C         SOLUTION OF HEAT CONDUCTION EQUATIONS IN CONTRL. THE POSITION  018653
C         OF THERMAL EQUILIBRIUM IS THE POSITION THAT ELASTIC RESTORING  018654
C         FORCES ARE MEASURED FROM. THIS COMPUTATION IS CARRIED OUT      018655
C         IN GMISC. THIS IS AN ADVANCED TOPIC; IT SUFFICES TO SAY THAT   018656
C         GMISC CAN BE SETUP TO INCLUDE THIS EFFECT IF REQUIRED.         018657
C                                                                     018658
C                                                                     018659
C         SUBROUTINE EQADD   SENSOR AND TORQUE SIGNAL EQUATIONS          018660
C         ----------------------------------------------------           018661
C                                                                     018662
C     THE CONTROL SYSTEM DESIGNER NEEDS THE ABILITY TO OBTAIN THE        018663
C         TRANSFER FUNCTIONS WHICH RELATE PHYSICALLY REALIZABLE SENSOR   018664
C         SIGNALS, WHICH ARE FUNCTIONS OF PLANT STATE VARIABLES, TO      018665
```

```
C        PHYSICALLY REALIZABLE TORQUE SIGNALS, WHICH ARE FUNCTIONS       018666
C        OF CONTROL SYSTEM STATE VARIABLES.                              018667
C                                                                        018668
C     TO ACCOMPLISH THIS THE USER MUST DEFINE A SET OF NAUX AUXILARY     018669
C        EQUATIONS. I.E. SUBROUTINE CONTRL THE USER DEFINED THAT IN      018670
C        SUBROUTINE EQADD THERE WOULD BE CODED EXACTLY                   018671
C            NXSS  SENSOR SIGNAL EQUATIONS, AND EXACTLY                  018672
C            NBTQ  TORQUE SIGNAL EQUATIONS                               018673
C        SUCH THAT                                                       018674
C                     NAUX = NXSS+NBTQ                                   018675
C                                                                        018676
C     AS IN ALL OTHER USER DEFINED ROUTINES CERTIAN RULES MUST BE        018677
C        ADHERED TO BY THE USER. THESE ARE AS FOLLOWS:                   018678
C         1) SENSOR SIGNALS - THE USER MUST DEFINE NXSS SENSOR SIGNAL    018679
C                             EQUATIONS. THESE EQUATIONS ARE ASSUMED     018680
C                             TO BE FUNCTONS OF PLANT STATE VARIABLES.   018681
C                             IN THE PROCESS OF NUMERICALLY LINEARIZING  018682
C                             EQUATIONS ALL SENSOR SIGNAL EQUATIONS      018683
C                             WILL BE EXPRESSED AS LINEAR FUNCTIONS OF   018684
C                             PLANT STATE VARIABLES                      018685
C              THE USER CODES IN EQADD                                   018686
C                 YDT(NEQ+I) = I-TH SENSOR SIGNAL EQUATION               018687
C                                   I=1,2,....NXSS                       018688
C         2) TORQUE SIGNALS - THE USER MUST DEFINE NBTQ TORQUE SIGNAL    018689
C                             EQUATIONS. THESE EQUATIONS ARE ASSUMED     018690
C                             TO BE FUNCITONS OF CONTROL SYSTEM STATE    018691
C                             VARIABLES. THE USER MUST BE CAREFUL TO     018692
C                             INCLUDE CONTROL EQUATIONS IN CONTRL WHICH  018693
C                             CAN BE USED TO DEFINE ALL TORQUE SIGNALS.  018694
C                             THIS IS NECESSARY BECAUSE IN THE           018695
C                             NUMERICAL LINEARIZATION PROCESS ALL        018696
C                             TORQUE SIGNALS WILL BE EXPRESSED AS        018697
C                             LINEAR FUNCTIONS OF THE CONTROL SYSTEM     018698
C                             STATE VARIABLES                            018699
C               THE USER CODES IN EQADD                                  018700
C                 YDT(NEQ+NXSS+J) = J-TH TORQUE SIGNAL EQUATION          018701
C                                      J=1,2,....NBTQ                    018702
C                                                                        018703
C                                                                        018704
C     SYSTEM AND BODY ENERGY AND MOMENTUM                                018705
C-------------------------------------                                   018706
C                                                                        018707
C     THE FOLLOWING PARAMETERS ARE CONTINUALLY MONITORED FOR THE USER    018708
C        TO CHECK TO DETERMINE IF ALL ENERGY AND MOMENTUM CONSERVATION   018709
C        LAWS ARE HOLDING TRUE. THESE PARAMETERS ARE WITH THE EXCEPTION  018710
C        OF ORDINARY MOMENTA COMPUTED IN SUBROUTINE ENGMOM, THEY ARE     018711
C        ALL STORED IN ZMOMENGZ AND PRINTED AS A FUNCTION OF TIME        018712
C                                                                        018713
C     P        = ORDINARY MOMENTA ARRAY, SEE EQUATION II-49 VOL I        018714
C                DIMENSIONED (NBMAX*(6+NMDHCO)+NMMAX)                    018715
C        (P(I),I=1,NU)  = ORDINARY MOMENTA ASSOCIATED WITH VELOCITY      018716
C                          COORDINATE I, I=1,....NU                      018717
C     PMOM     = BODY ANGULAR AND LINEAR MOMENTUM ARRAY                  018718
C                DIMENSIONED (6*NBMAX)                                   018719
C        (PMOM(I),I=6*(N-1)+1,6*(N-1)+3) = 3 COMPONENTS OF THE INERTIAL  018720
C                          ANGULAR MOMENTUM VECTOR ASSOCIATED WITH THE   018721
C                          MOTION OF BODY N WRT INERTIAL COORDINATES     018722
C        (PMOM(I),I=6*(N-1)+4,6*(N-1)+6) = 3 COMPONENTS OF THE INERTIAL  018723
C                          LINEAR MOMENTUM VECTOR ASSOCIATED WITH THE    018724
C                          MOTION OF BODY N WRT INERTIAL COORDINATES     018725
```

```
C       HTOT   = TOTAL SYSTEM ANGULAR MOMENTUM ARRAY DIMENSIONED (3) THE 3
C                COMPONENTS ARE WRT INERTIAL COORDINATES
C       PTOT   = TOTAL SYSTEM LINEAR MOMENTUM ARRAY DIMENSIONED (3) THE 3
C                COMPONENTS ARE WRT INERTIAL COORDINATES
C       ENGKE  = KINETIC ENERGY ARRAY, DIMENSIONED (NBMAX)
C                ENGKE(N) =KINETIC ENERGY OF BODY N
C       ENGPE  = POTENTIAL ENERGY ARRAY, DIMENSIONED (NBMAX)
C                ENGPE(N) = POTENTIAL ENERGY OF BODY N
C       TOTKE  = TOAL SYSTEM KINETIC ENERGY
C       TOTPE  = TOTAL SYSTEM POTENTIAL ENERGY
C       PTOTL  = TOTAL SYSTEM LINEAR MOMENTUM (VECTOR MAGNITUDE)
C       AHTOT  = TOTAL SYSTEM ANGULAR MOMENTUM (VECTOR MAGNITUDE)
C
C
C           LINEAR - LINEARIZED SYSTEM EQUATIONS OF MOTION
C-----------------------------------------------------------------
C
C       THE LINEARIZATION PROCESS PROVIDES A SET OF LINEAR COUPLED FIRST
C           ORDER DIFFERENTIAL EQUATIONS WHICH DEFINE THE TOTAL SYSTEM
C           RESPONSE RELATIVE TO AN EQUILIBRIUM STATE.
C
C       IF THE COUPLED NON-LINEAR EQUATIONS OF MOTION ARE TO BE LINEARIZED
C           THE INITIAL STATE DEFINED BY THE INPUTTED SYSTEM INITIAL
C           CONDITIONS DEFINES THE EQUILIBRIUM STATE RELATIVE TO WHICH
C           PERTURBATION EQUATIONS ARE TO BE DERIVED
C
C             LET:
C           (Y(I,0),I=1,NEQ)      = EQUILIBRIUM STATE OF THE STATE VECTOR
C                                   DEFINED BY USER SPECIFIED INITIAL
C                                   CONDITIONS
C           (Y(I,T),I=1,NEQ)      = STATE VECTOR AT TIME T
C            THEN:
C
C                (DELY(I,T) = Y(I,T)-Y(I,0),I=1,NEQ)
C
C            IS THE PERTURBATION VECTOR
C
C             SIMILARLY LET:
C           (YDT(I,0),I=1,NEQ)  = EQUILIBRIUM STATE OF THE FIRST TIME
C                                 DERIVATIVE OF THE STATE VECTOR
C           (YDT(I,T),I=1,NEQ)  = STATE RATE VECTOR AT TIME T
C           (YDT(I,0),I=NEQ+1,NAUX) = EQUILIBRIUM STATE FOR ALL AUXILARY
C                                     EQUATIONS DEFINED BY THE USER IN
C                                     LOADD, THESE ARE SENSOR AND TORQUE
C                                     SIGNAL EQUATIONS FOR TRANSFER
C                                     FUNCTION STUDIES
C           (YDT(I,T),I=NEQ+1,NAUX) =  SENSOR AND TORQUE SIGNALS AT TIME T
C            THEN:
C
C                (DELYDT(I,T) = YDT(I,T)-YDT(I,0),I=1,NEQ+NAUX)
C
C            IS THE PERTURBATION RATE VECTOR
C
C       THE NUMERICALLY OBTAINED LINEARIZED PERTURBATION EQUATIONS
C           TO BE OBTAINED VIA SUBROUTINE LINEAR ARE:
C
C                    DELYDT(I,T) = H(I,J)*DELY(J,T) ;  I=1,2,....,NEQ+NAUX
C
C           WHERE SUMMATION IS IMPLIED BY THE REPEATED J INDEX
C                         J=1,2,....,NEQ
```

```
C          THE ELEMENTS OF THE NEQ+NAUX BY NEQ  PARTITIAL DERIVATIVE
C          MATRIX 'H' ARE OBTAINED A COLUMN AT A TIME
C
C      ONLY THOSE COLUMNS J OF THE PARTIAL DERIVATIVE MATRIX 'H'
C          CORRESPONDING TO INDEPENDENT COORDINATES ARE REQUIRED; THAT IS
C          ONLY THOSE COLUMNS J FOR WHICH INDEP(J)=1, J=1,2,...,NEQ
C          ARE EVALUATED. A MAXIMUM OF 30 ITERATION STEPS ARE ALLOWED
C          FOR THE CONVERGENCE OF EACH COLUMN. IF DEBUG(40)=.TRUE. THE
C          FOLLOWING PRINTOUT IS OBTAINED
C      ' Z     ARRAY ITERATION N FOR COLUMN J ' = THE NEQ+NAUX ELEMENTS IN
C                         COLUMN J OF THE PARTIAL DERIVATIVE MATRIX AT
C                         THE START OF ITERATION N
C      ' ZNEW ARRAY ITERATION N FOR COLUMN J ' = THE NEQ+NAUX ELEMENTS IN
C                         COLUMN J OF THE PARTIAL DERIVATIVE MATRIX AT
C                         THE END OF ITERATION N
C           FOR CONVERGENE THE FOLLOWING CODED TEST IS MADE
C                  DO 100 I=1,NEQ+NAUX
C                  IF(INDEP(I).EQ.0) GO TO 100
C                  DN  = DABS(Z(I))
C                  DN1 = DABS(ZNEW(I))
C                  IF(DN1.GT.DN)   DN=DN1
C                  IF(DN .LT.EPS1) GO TO 100
C                  G1  = DABS(ZNEW(I)-Z(I))/DN
C                  IF(G1 .LE.EPS2) GO TO 100
C           C       SORRY TRY AGAIN; ONLY 30 CHANCES ALLOWED
C                   TO ACHIEVE CONVERGENCE FOR ALL COMPONENTS
C               100 CONTINUE
C           C       CONVERGENCE TEST SUCCESSFUL
C           C       EPS1 = 1.0D-08      THESE NUMBERS MAY BE PROBLEM
C           C       EPS2 = 1.0D-04      SENSITIVE
C           ** THIS CRITERIA HAS BEEN SLIGHTLY MODIFIED TO PROVIDE
C              FOR TWO AUTOMATIC CHANGES OF EPS1. FOR COLUMNS WITH VERY
C              LARGE MAGNITUDE NUMBERS EPS1 = 1.0D-08 IS BEYOND THE
C              ACCURACY CAPABILITIES OF THE COMPUTER. THE CRITERIA FOR
C              RESETTING IS BASED UPON THE LARGEST MAGNITUDE NUMBER
C              IN THE Z ARRAY FOUND ON THE FIRST ITERATION THROUGH.
C                IF EPS1 = 1.0D-08 IS UNSUCCESSFUL
C              RESET     EPS1 = 1.0D-06*MAX(Z(I) OF ITERATION 1) AND
C              THEN TO 1.0D-04*MAX(Z(I) OF ITERATION 1) BEFORE GIVING UP
C
C          WHEN SUCCESSFUL CONVERGENCE OBTAINED THE FOLLOWING PRINTOUT
C          IS PROVIDED
C      ' COLUMN J OF THE PARTIAL DERIVATIVE MATRIX IS' = THE NJG ELEMENTS
C                         IN COLUMN J OF THE PARTIAL DERIVATIVE MATRIX
C                         WHICH MUST BE SAVED FOR LINEAR ANALYSIS. THESE
C                         ARE THE ELEMENTS OF ARRAY ZNEW CORRESPONDING TO
C                         EITHER INDEPENDENT COORDINATES OR SENSOR AND
C                         TORQUE SIGNALS. (THIS COLUMN MATRIX IS STORED
C                         ON NTAPE2 FOR RETRIEVAL IN DYNS40
C                         NOTE:  NJG = NX + NAUX
C
C
C
C
C
C
C               STABILITY ANALYSIS OF LINEARIZED EQUATIONS, DYNS40
C               **************************************************
C
C        ** NOTE **    STABILITY ANALYSIS OF LARGE ORDER SYSTEMS, WHICH
```

```
C                    IS WHAT DISCUS IS GEARED FOR. REQUIRES OPERATING
C                    WITH LARGE ORDER MATRICES. TO CONSERVE MEMORY
C                    REQUIREMENTS. WORK AREAS AND SCRACH FILES ARE
C                    EXTENSIVELY USED. BY SETTING THE APPROPRIATE
C                    DEBUG FLAG EQUAL .TRUE., PARTICULAR MARICES CAN
C                    BE PRINTED. IN MOST CASES THEY ARE DESTROYED AS
C                    SOON AS THEY ARE NO LONGER NEEDED.
C
C
C         LINEARIZED EQUATIONS OF MOTION
C         ------------------------------
C
C      THE STABILITY ANALYSIS IS DIRECTED BY SUBROUTINE DYNS40. IN
C         SUBROUTINE LINEAR THE COUPLED SYSTEM EQUATIONS ALONG WITH THE
C         SENSOR AND TORQUE SIGNAL EQUATIONS ARE LINEARIZED AND THE
C         PARTIAL DERIVATIVE MATRIX IS STORED FOR RECOVERY IN DYNS40 ON
C         NTAPE2. UPON ENTRY INTO DYNS40 THIS MATRIX IS READ, STORED
C         AND PRINTED. THE EQUATIONS ARE OF THE FORM
C
C                DLYDT(I,T) = H(I,J)*DLY(J,T)
C
C         WHERE:
C         I=1,....,NX-NDELTA   INDICES ASSOCIATED WITH INDEPENDENT PLANT
C                              COORDINATES
C         I=NX-NDELTA+1,....,NX INDICES ASSOCIATED WITH INDEPENDENT
C                              CONTROL SYSTEM COORDINATES
C         I=NX+1,....,NX+NASS  INDICES ASSOCIATED WITH SENSOR SIGNALS
C                              AS DEFINED IN EQASS
C         I=NX+NASS+1,....,NJG INDICES ASSOCIATED WITH TORQUE SIGNALS
C                              AS DEFINED IN EQASS
C         AND THE REPEATED INDEX J IMPLIES SUMMATION OVER J;J=1,....,NX
C         WORK ARRAYS W1 AND W2 IN /SCRCH1/ ARE USE AS TEMPORARY STORAGE
C         LOCATIONS FOR PARTITIONS OF THE H MATRIX. THE SYMBOLS DLY AND
C         DLYDT ARE DEFINED ABOVE UNDER THE TITLE 'LINEAR - ...'
C                                        -
C      THE OUTPUT MATRICES PRINTED AFTER THE FOLLOWING CAPTIONS ARE:
C       * OUTPUT MATRIX -A-      (NX X NX )' = ((H(I,J),J=1,NX),I=1,NX);
C                       THAT IS, THE COEFFICIENTS OF THE PARTIAL
C                       DERIVATIVE MATRIX ASSOCIATED WITH THE
C                       INDEPENDENT PLANT AND CONTROL SYSTEM STATE
C                       VARIABLES (THESE ARE DEFINED BY ARRAY INDEF)
C       * OUTPUT MATRIX -A-AUX (NAUX X NX)' = ((H(I,J),J=1,NX),I=1,NAUX);
C                       THAT IS, THE COEFFICIENTS OF THE PARTIAL
C                       DERIVATIVE MATRIX ASSOCIATED WITH THE EQUATIONS
C                       WHICH DEFINE SENSOR AND TORQUE SIGNALS AS
C                       LINEAR FUNCTIONS OF PLANT AND CONTROL SYSTEM
C                       STATE VARIABLES RESP.
C
C          ** NOTE **     THE CONTENTS OF THIS MATRIX ARE PARTICULARLY
C                    USEFUL WHEN THE HIGHLY COUPLED SYSTEMS ARE
C                    MODELLED. THE RELATIVE MAGNITUDES OF THE ELEMENTS
C                    IN ANY ROW PROVIDE A MEASURE OF THE PARTICULAR
C                    SENSOR OR TORQUE SIGNAL'S DEPENDENCE ON EACH OF
C                    THE SYSTEM'S INDEPENDENT DEGREES OF FREEDOM IN
C                    THE TIME DOMAIN
C
C
C         SIMILARITY TRANSFORMATION
C         -------------------------
C
```

```
C     TO OBTAIN TRANSFER FUNCTIONS BETWEEN SELECTED SENSOR SIGNALS AND    018906
C        SELECTED TORQUE SIGNALS A SIMILARITY TRANSFORMATION IS MADE      018907
C        WHICH NUMERICALLY SELECTS IN AN OPTIMUM MANNER EXACTLY NAUX      018908
C        INDEPENDENT SYSTEM STATE VARIABLES FROM THE STATE VECTOR        018909
C        AND REPLACES THEM WITH THE NAUX SENSOR AND TORQUE SIGNAL         018910
C        VARIABLES. THE RESULTING EQUATION III-24 OF VOL I IS OF THE      018911
C        FORM: (THIS OPERATION IS DONE IN ASIMLR AND FINDT)               018912
C                                                                         018913
C                    ZDT(I,T) = R**(-1)*H*R(I,J) * Z(J,T)                 018914
C                                                                         018915
C        WHERE:                                                           018916
C        (I,I=1,NY2)         = INDICES OF THE PERTURBATION VARIABLES      018917
C                              ASSOCIATED WITH THE PLANT WHICH ARE NOT    018918
C                              ELIMINATED BY THE SIMILARITY               018919
C                              TRANSFORMATION                             018920
C        (I,I=NY2+1,         = INDICES OF THE PERTURBATION VARIABLES      018921
C            NY2+NKSS)         ASSOCIATED WITH THE NKSS SENSOR SIGNAL     018922
C                              VARIABLES                                  018923
C        (I,I=NX-NDELTA,     = INDICES OF THE PERTURBATION VARIABLES      018924
C             NX-NSTQ)         ASSOCIATED WITH THE CONTROL SYSTEM WHICH   018925
C                              ARE NOT ELIMINATED BY THE SIMILARITY       018926
C                              TRANSFORMATION                             018927
C        (I,I=NX-NSTQ+1,     = INDICES OF THE PERTURBATION VARIABLES      018928
C            NX)               ASSOCIATED WITH THE NSTQ TORQUE SIGNAL     018929
C                              VARIABLES                                  018930
C        Z(I,T)              = PERTURBATION VARIABLE I AT TIME T REFER    018931
C                              TO 'THE TRANSFORMED STATE VECTOR           018932
C                              CORRELATION ARRAY' PRINTED TO CORRELATE    018933
C                              INDEX I WITH ORIGINAL STATE VARIABLE       018934
C                              INDEXING STORED IN LEXO AND LOCO           018935
C        ZDT(I,T)            = FIRST TIME DERIVATIVE OF Z(I,T)            018936
C                              (INDEXING CONSISTANT WITH Z(I,T))          018937
C        R**(-1)*H*R(I,J)    = THE (I,J) ELEMENT OF THE SIMILARITY        018938
C                              TRANSFORMATION MATRIX (OUTPUT OF ASIMLR)   018939
C                              ORDERED AS JUST DEFINED ABOVE              018940
C                                                                         018941
C     THE OUTPUT MATRICES PRINTED AFTER THE FOLLOWING CAPTIONS ARE        018942
C      'OUTPUT MATRIX -A*-     (NX X NX)    = (R**(-1)*H*R(I,J),J=1,NX),  018943
C                       I=1,NX); THAT IS THE ELEMENTS OF THE SIMILARITY   018944
C                       TRANSFORMATION MATRIX COMPUTED IN ASIMLR AND      018945
C                       PASSED BACK TO DYNS40 IN THE ARRAY W2             018946
C          'ETA'                             = A LISTING IN DECREASING    018947
C      'NO  REAL PART IMAGINARY PART'          ORDER OF REAL PART OF ALL  018948
C                                              NX COMPLEX ROOTS OF THE    018949
C                                              PARTIAL DERIVATIVE MATRIX  018950
C                                              'H' DEFINED ABOVE          018951
C          'ETA*'                            = A LISTING IN DECREASING    018952
C      'NO  REAL PART IMAGINARY PART'          ORDER OF REAL PART OF ALL  018953
C                                              NX COMPLEX ROOTS OF THE    018954
C                                              SIMILARITY TRANSFORMATION  018955
C                                              MATRIX R**(-1)*H*R DEFINED 018956
C                                              ABOVE                      018957
C            ** NOTE **                                                   018958
C       BOTH SETS OF COMPLEX ROOTS                                        018959
C       SHOULD BE EQUAL. THIS IS A                                        018960
C       CHECK ON THE QUALITY OF THE                                       018961
C       SIMILARITY TRANSFORMATION                                         018962
C                                                                         018963
C                                                                         018964
C         COMPLEX ROOTS VIA THE QR-ALGORITHM                              018965
```

```
C          ----------------------------------
C
C      COMPLEX ROOTS OF GENERAL NON-SYMMETRIC MATRICES OF REAL NUMBERS
C          ARE OBTAINED VIA THE QR-ALGORITHM. QRDRVR ACCEPTS THE N BY N
C          MATRIX G AND RETURNS TO THE CALLING ROUTINE THE COMPLEX ROOTS
C          IN THE ARRAYS RR (REAL ROOTS) AND RI (IMAGINARY ROOTS). THE
C          ROOTS ARE ARRANGED IN DECREASING ORDER OF REAL PART.
C
C      IF DEBUG(64)   .TRUE. THE FOLLOWING CAPTIONS AND ASSOCIATED
C          MATRICES WILL BE PRINTED:
C       ' ELEMENTS OF ARRAY G'   = THE N BY N INPUT MATRIX FOR WHICH THE
C                                  COMPLEX ROOTS ARE TO BE OBTAINED
C       ' UPPER HESSENBERG FORM  = THE G-MATRIX IS PASSED TO SUBROUTINE
C         OF G OUT OF SUBQIA IS'   SUBQIA IN WHICH IT IS TRANSFORMED INTO
C                                  UPPER HESSENBERG FORM (ALMOST
C                                  TRIANGULAR) IN SO DOING THE ORIGINAL
C                                  MATRIX IS DESTROYED. DONE TO INCREASE
C                                  SPEED AND EFFICIENCY OF QR-ALGORITHM
C       ' REAL ROOTS OUT OF      = THE REAL PARTS OF THE N COMPLEX ROOTS
C         QRCCN ARE'               OF THE INPUT MATRIX G IN DECREASING
C                                  ORDER
C       ' IMAGINARY ROOTS OUT OF = THE ASSOCIATED IMAGINARY PARTS OF THE
C         QRCCN ARE'               N COMPLEX ROOTS, RESPECTIVELY
C
C
C          ASIMER & FINDT - PERFORM THE SIMILARITY TRANSFORMATION
C          ------------------------------------------------------
C
C      DISCOS HAS THE UNIQUE FEATURE THAT TRANSFER FUNCTIONS MAY BE
C          OBTAINED BETWEEN PHYSICALLY REALIZABLE INPUT SENSOR SIGNALS
C          AND PHYSICALLY REALIZABLE TORQUE SIGNALS. THE SENSOR AND
C          TORQUE SIGNALS ARE USER DEFINED FUNCTIONS SPECIFIED IN EQADD.
C          TO ACHIEVE THIS CAPABILITY A SIMILARITY TRANSFORMATION MUST
C          BE DEVELOPED WHICH WILL ALLOW FOR AN EXCHANGE OF VARIABLES;
C          THAT IS, NXSS PLANT STATE VARIABLES ARE ELIMINATED FROM THE
C          STATE VECTOR AND REPLACED BY NXSS SENSOR SIGNAL VARIABLES.
C          SIMILARLY NBTQ CONTROL STATE VARIABLES ARE ELIMINATED AND
C          REPLACED WITH NBTQ TORQUE SIGNAL VARIABLES . (SEE PAGE III-7
C          OF VOL I). THE MATRICES COMPUTED IN THESE ROUTINES ARE ONLY
C          STORED TEMPORARILY IN THE WORK ARRAYS W1 AND W2 IN /LWORK1/
C
C      BY SETTING DEBUG(65) AND DEBUG(66)=.TRUE. THE STEPS TAKEN THROUGH
C          ASIMER AND FINDT TO ARRIVE AT THE SIMILARITY TRANSFORMATION
C          CAN BE FOLLOWED FOR ANY PARTICULAR PROBLEM. THE FOLLOWING
C          CAPTIONS AND ASSOCIATED MATRICES ARE PRINTED:
C      ' INPUT MATRIX COEFFICIENTS  = THE NAUX BY NX MATRIX DERIVED IN
C        FOR LINEARIZED SIGNALS'      LINEAR WHICH CAN BE USED TO DEFINE
C                                     ALL SENSOR AND TORQUE SIGNALS AS
C                                     LINEAR FUNCTIONS OF INDEPENDENT
C                                     PLANT AND CONTROL SYSTEM STATE
C                                     VARIABLES. THIS IS THE INPUT MATRIX
C                                     TO ASIMER. IT IS THE PARTITION 'C'
C                                     IN EQUATION III-26
C      ' SUBROUTINE FINDT INPUT     = THE NAUX BY NJJ COEFFICIENT MATRIX
C        MATRIX'                      IN EQUATION III-26. FINDT USES THIS
C                                     MATRIX TO FIND THE BEST SET OF
C                                     STATE VARIABLES TO BE REPLACED BY
C                                     SENSOR AND TORQUE SIGNAL VARIABLES
C      ' VARIABLE EXCHANGE MATRIX'  = THE NAUX BY NJJ VARIABLE EXCHANGE
C                                     MATRIX ILLUSTRATED PAGE E3.
```

```
C                                           VOL 1. THE EXISTANCE OF 1.0000D+00     019026
C                                           IN COLUMN J OF ROW I IMPLIES THAT      019027
C                                           THE J-TH INDEPENDENT STATE VARIABLE    019028
C                                           WILL BE NOT SURVIVE THE EXCHANGE       019029
C                                           PROCESS. THE STATE VARIABLES TO BE     019030
C                                           DELETED ARE THOSE WHICH THE SENSOR     019031
C                                           AND TORQUE SIGNALS ARE MOST            019032
C                                           STRONGLY DEPENDENT UPON. THESE ARE     019033
C                                           DETERMINED NUMERICALLY BY AN           019034
C                                           ALGORITHM WHICH ASSUMES THAT THE       019035
C                                           SIGNALS ARE MUTUALLY INDEPENDENT.      019036
C     ' SIMILARITY TRANSFORMATION  = THE NX BY NX SIMILARITY                       019037
C       MATRIX IS'                          TRANSFORMATION MATRIX. IT IS           019038
C                                           CONSTRUCTED DIRECTLY FROM THE          019039
C                                           VARIABLE EXCHANGE MATRIX AND           019040
C                                           RETURNED TO ASIMLR IN THE DUMMY        019041
C                                           ARRAY 'T' DEFINED IN FINDT. IT IS      019042
C                                           THE MATRIX 'R' IN EQUATION III-23      019043
C                                           OF VOL I BUT NOT ORDERED EXACTLY       019044
C                                           AS SHOWN THEREIN                       019045
C        LET:                                                                      019046
C             YY  = THE NX BY 1 COLUMN MATRIX OF INDEPENDENT PLANT AND             019047
C                   CONTROL SYSTEM STATE VARIABLES (INCREASING INDEX               019048
C                   ORDER)                                                         019049
C             Y*  = THE NX BY 1 COLUMN MATRIX OF INDEPENDENT PLANT AND             019050
C                   CONTROL SYSTEM STATE VARIABLES TO BE RETAINED (THE             019051
C                   FIRST NX-NAUX ROWS) AND THE SENSOR AND TORQUE                  019052
C                   SIGNAL VARIABLES (LAST NAUX ROWS)                              019053
C             R   = THE NX BY NX SIMILARITY TRANSFORMATION MATRIX 'R'              019054
C        THEN                                                                      019055
C                         YY(I) = R(I.J) * Y*(J)                                   019056
C        WHERE I 1 ....NX AND REPEATED INDEX J IMPLIES SUMMATION                   019057
C                                                                                  019058
C     ' THE TRANSFORMED STATE      = NX BY 1 ARRAY COMPUTED IN FINDT AND           019059
C       VECTOR CORRELATION ARRAY     PRINTED THEREIN. IT IS STORED ONLY            019060
C       FOLLOWS' = IV                TEMPORARILY IN IV IN /LIJV/. THE              019061
C                                    PRINTED ARRAY ELEMENTS HAVE THE               019062
C                                    FOLLOWING MEANING: ALL INDICIES               019063
C                                    ARE CONSISTANT WITH THE LENO AND              019064
C                                    LOCO ARRAYS                                   019065
C          (IV(I).I=1.NY2)         = INDICES ASSOCIATED WITH THE PLANT             019066
C                                    STATE VARIABLES WHICH ARE NOT                 019067
C                                    ELIMINATED BY THE SIMILARITY                  019068
C                                    TRANSFORMATION                                019069
C          (IV(I).I=NY2+1.         = INDICES ASSOCIATED WITH THE SENSOR            019070
C                    NY2+NXSS)       SIGNAL VARIABLES WHICH ARE TO BE              019071
C                                    USED FOR TRANSFER FUNCTION STUDIES            019072
C          (IV(I).I=NY2+NASS+1.    = INDICES ASSOCIATED WITH THE CONTROL           019073
C                    NX-NBIG         SYSTEM STATE VARIABLES WHICH ARE              019074
C                                    NOT ELIMINATED BY THE SIMILARITY              019075
C                                    TRANSFORMATION                                019076
C          (IV(I).I=NX-NBIG+1.NX) = INDICES ASSOCIATED WITH THE TORQUE             019077
C                                    SIGNAL VARIABLES WHICH ARE TO BE              019078
C                                    USED FOR TRANSFER FUNCTION STUDIES            019079
C     ' OUTPUT MATRIX -T- '     = THE NX BY NX SIMILARITY TRANSFORMATION           019080
C                                 MATRIX RETURNED TO ASIMLR IN THE DUMMY           019081
C                                 ARRAY T FROM FINDT.                              019082
C     ' OUTPUT MATRIX Y*'       = THE NX BY 1 COLUMN MATRIX OF RETAINED            019083
C                                 INDEPENDENT SYSTEM STATE VARIABLES AND           019084
C                                 SENSOR AND TORQUE SIGNAL VARIABLES AS            019085
```

```
C                                 DEFINED ABOVE.                              019086
C                                                                             019087
C        LET                                                                  019088
C                  B = T**(-1)    (MATRIX OPERATION)                          019089
C        THEN                                                                 019090
C                  Y*(I) = B(I,J)*YY(J)                                       019091
C                                                                             019092
C        WHERE YY IS DEFINED ABOVE AND SUMMATION IMPLIED BY REPEATED J        019093
C                                                                             019094
C     'R**(-1)*H*R MATRIX IS'  = THE NX BY NX MATRIX WHICH IS THE             019095
C                                TRANSFORM OF THE PARTIAL DERIVATIVE          019096
C                                MATRIX 'H' COMPUTED IN LINEAR, BY THE        019097
C                                SIMILARITY TRANSFORMATION MATRIX 'R'         019098
C                                COMPUTED IN FINOUT. THIS IS THE MATRIX       019099
C                                R**(-1)*H*R APPEARING IN EQUATION            019100
C                                III-24 OF VOL I; THE ROWS AND COLUMNS        019101
C                                HOWEVER AT THIS POINT ARE CONSISTANT         019102
C                                WITH Y* COLUMN MATRIX DEFINED ABOVE          019103
C     'REORDERED R**(-1)*H*R   = THE NX BY NX MATRIX R**(-1)*H*R WHICH        019104
C      MATRIX IS'                HAS HAD ITS ROWS AND COLUMNS                 019105
C                                INTECHANGED SO AS TO BE CONSISTANT           019106
C                                WITH THE TRANSFORMED STATE VECTOR            019107
C                                CORRELATION ARRAY PRINTED AND DEFINED        019108
C                                ABOVE. THIS IS THE MATRIX WHICH IS           019109
C                                RETURNED TO DYNS43. IT IS THE                019110
C                                DESIRED SIMILARITY TRANSFORMATION            019111
C                                MATRIX (NOT TO BE CONFUSED WITH THE          019112
C                                MATRIX R ALONE)                              019113
C                                                                             019114
C                                                                             019115
C     FREQUENCY DOMAIN ANALYSIS                                               019116
C---------------------------------                                            019117
C                                                                             019118
C        DESCRIPTION OF TRANSFER FUNCTION   (THE PARAMETERS WHICH WILL        019119
C        --------------------------------    DEFINE WHAT TRANSFER             019120
C                                            FUNCTIONS ARE TO BE SETUP        019121
C                                            AND HOW THE FREQUENCY            019122
C                                            RESPONSE DATA IS TO BE           019123
C                                            DISPLAYED MUST BE USER           019124
C                                            SPECIFIED ON INPUT CARDS         019125
C                                            TO BE READ IN DYNS40)            019126
C     LNAM    = INPUT PARAMETER, STORED LOCALLY                               019127
C             IF(LNAM.EQ.4HTIME) LINEAR TIME RESPONSE ANALYSIS                019128
C                                REQUESTED BY USER                            019129
C             IF(LNAM.EQ.4HFREQ) FREQUENCY ANALYSIS REQUESTED BY USER         019130
C             IF(LNAM.EQ.4H    ) FURTHER ANALYSIS OF LINEARIZED               019131
C                                EQUATIONS NOT REQUESTED                      019132
C     NCYC    = TOTAL NUMBER OF TRANSFER FUNCTIONS WHICH WILL BE SETUP        019133
C             AND ANALYSED IN THE FREQUENCY DOMAIN. INPUT PARAMETER           019134
C             STORED LOCALLY. (MAXIMUM PER RUN EQUALS 11)                     019135
C     LRY     = TRANSFER FUNCTION DATA SPECIFICATION ARRAY. STORED            019136
C             LOCALLY AND DIMENSIONED (9,11) TO AVOID EXCESSIVE USE           019137
C             OF SUBSCRIPTED VARIABLES THE FOLLOWING PARAMETERS ARE           019138
C             SET, FOR TRANSFER FUNCTION ICYC,ICYC=1,2,...,NCYC               019139
C        ITYPE  = LRY(1,ICYC) = TRANSFER FUNCTION TYPE, SEE SUBROUTINE        019140
C                 TFTYPE FOR DESCRIPTION OF 8 POSSIBLE TYPES AVAILABLE        019141
C                 FOR USER                                                    019142
C        ITFIN  = LRY(2,ICYC) = INTEGER USED TO SPECIFY THE INPUT             019143
C                 SIGNAL ASSOCIATED WITH THE TRANSFER FUNCTION                019144
C        ITFOUT = LRY(3,ICYC) = INTEGER USED TO SPECIFY THE OUTPUT            019145
```

```
C                  SIGNAL ASSOCIATED WITH THE TRANSFER FUNCTION              019146
C         KPLOT  = IFY(4,ICYC) =INTEGER USED TO DEFINE IF PLOTS TO BE        019147
C                  GENERATED                                                 019148
C         IAFLG  = IFY(5,ICYC) = INTEGER USED TO DEFINE IF CHARACTERISTIC    019149
C                  MATRIX OR ITS TRANSPOSE TO BE USED IN ANALYSIS            019150
C         NFKP   = IFY(6,ICYC) IF ITYPE.NE.7 UNUSED OTHERWISE = NUMBER OF    019151
C                  FEEDBACK SIGNALS TO RETAIN (MAX OF 3 ALLOWED)             019152
C         KFKP(I)= IFY(6+I,ICYC),I=1,NFKP = INTEGERS DEFINING WHICH          019153
C                  FEEDBACK SIGNALS ARE TO BE RETAINED                       019154
C      IRY       = TRANSFER FUNCTION COMPUTATIONAL TOLERANCE DATA ARRAY.     019155
C                STORED LOCALLY AND DIMENSIONED (3,LI). FOR SOME SPECIAL     019156
C                PROBLEMS DEFAULT TOLERANCE DATA WILL NOT WORK. THE DATA     019157
C                IN THIS ARRAY ALLOWS FOR OVERRIDE CAPABILITY                019158
C         TOLN   = TOLD = 10.D0 ** IRY(1,ICYC) = ROOT TOLERANCE FACTOR       019159
C                ALL NUMERATOR AND DENOMINATOR ROOTS LESS THAN THIS FACTOR   019160
C                SET EQUAL TO ZERO IN SIFT. REAL*8 CONSTANT STORED IN        019161
C                ZETOLZ                                                      019162
C         GTOL   = 10.D0 ** IRY(2,ICYC) = THE NUMERATOR GAIN MUST BE         019163
C                  GREAT THAN GTOL FOR COMPUTATION TO CONTINUE               019164
C         PTOL   = 10.D0 ** IRY(3,ICYC) =ALL ROOTS OF SECOND TERM IN         019165
C                  NUMERATOR OF TRANSFER FUNCTION LESS THAN PTOL SET         019166
C                  EQUAL TO ZERO (SEE NUMS AND EQ III-42). THIS REFERS       019167
C                  TO THE SPECIAL TECHNIQUE USED IN OBTAINING NUMERATOR      019168
C                  ROOTS                                                     019169
C                                                                            019170
C                                                                            019171
C                                                                            019172
C          TFTYPE - MATRICES FOR TRANSFER FUNCTION COMPUTATION               019173
C          ---------------------------------------------------               019174
C                                                                            019175
C      DISCOS HAS THE BUILT IN CAPABILITY TO GENERATE UPON USER SELECTION    019176
C          ANY ONE OF EIGHT DIFFERENT TRANSFER FUNCTIONS. THE BASIC          019177
C          MODEL IN COMPUTER MEMORY WHEN TFTYPE IS CALLED IS ILLUSTRATED     019178
C          BY THE FOLLOWING FIGURE                                           019179
C                                                                            019180
C                            ------------                                    019181
C            UI(S) +        |            |                                   019182
C          ---------->O---->| PLANT G(S) |------>O------>                    019183
C                    -|     |            |       |                           019184
C              UCS(S) |      ------------        |                           019185
C                     |                          |                           019186
C                     |      --------------      | XSS(S)                    019187
C                     |     |              |    +V                           019188
C                     O<----| CONTROL H(S) |<---O<-------                    019189
C                           |              |    + XS(S)                      019190
C                            --------------                                  019191
C                                                                            019192
C                                                                            019193
C          THE SIMULTANEOUS EQUATIONS WHICH HAVE BEEN DERIVED ARE IN         019194
C          THE S-DOMAIN GIVEN IN MATRIX NOTATION BY:                         019195
C                                                                            019196
C                  S*X(I,S) = A(I,J)*X(J,S) + B(I,K)*U(K,S)                  019197
C                                                                            019198
C          WHERE  I=1,2,....,NX                                              019199
C                 J=1,2,....,NX                                              019200
C                 K=1,2,....,NAUX     (NAUX=NXSS+NSTG)                       019201
C          AND SUMMATION IS IMPLIED BY REPEATED INDEX NOTATION.              019202
C                                                                            019203
C          FROM 'THE TRANSFORMED STATE VECTOR CORRELATION ARRAY':            019204
C           FOR INDEX III=1,....,NY2                                         019205
```

```
C              X(I,S) = Y(I,S)    = I-TH INDEPENDENT PERTURBATION VARIABLE  019206
C                                   FOR THE PLANT WHICH HAS NOT BEEN        019207
C                                   EXCHANGED BY THE SIMILARITY             019208
C                                   TRANSFORMATION FOR A SENSOR SIGNAL      019209
C           FOR INDEX I;I=NY2+1,....,NY2+NXSS                               019210
C              X(I,S) = XSS(L,S) = L-TH SENSOR SIGNAL (L=I-NY2) WHICH BY    019211
C                                   THE SIMILARITY TRANSFORMATION IS USED   019212
C                                   AS A PLANT STATE VARIABLE               019213
C           FOR INDEX I;I=NY +NXSS+1,....,NX-NBTQ                           019214
C              X(I,S) = Y(M,S)    = M-TH INDEPENDENT PERTURBATION VARIABLE  019215
C                                   FOR THE CONTROL SYSTEM (M=I-NY2-NXSS)   019216
C                                   WHICH HAS NOT BEEN EXCHANGED BY THE     019217
C                                   SIMILARITY TRANSFORMATION FOR A TORQUE  019218
C                                   SIGNAL                                  019219
C           FOR INDEX I;I=NX-NBTQ+1,....,NX                                 019220
C              X(I,S) = TQS(N,S) - N-TH TORQUE SIGNAL (N=I-NX+NBTQ) WHICH   019221
C                                   BY THE SIMILARITY TRANSFORMATION IS     019222
C                                   USED AS A CONTROL SYSTEM STATE          019223
C                                   VARIABLE                                019224
C           FOR INDICES I AND J; I=1,....,NX AND J=1,....,NX                019225
C              A(I,J) = R**(-1)*A*R(I,J) = (I,J) ELEMENT OF THE             019226
C                                   SIMILARITY TRANSFORMATION MATRIX        019227
C                                   RETURNED TO DYN340 FROM ASIMLR          019228
C           FOR INDEX K;K=1,NBTQ                                            019229
C              U(K,S) = RT(K,S)   = REFERENCE INPUT TORQUE SIGNAL THAT THE  019230
C                                   K-TH CONTROL TORQUE SIGNAL TQS(K,S) IS  019231
C                                   SUBTRACTED FROM AND FEED TO THE PLANT   019232
C           FOR INDEX K;K=NBTQ+1,....,NBTQ+NXSS                             019233
C              U(K,S) = RS(L,S)   = REFERENCE INPUT SENSOR SIGNAL WHICH IS  019234
C                                   ADDED TO THE L-TH SENSOR SIGNAL         019235
C                                   XSS(L,S) AND FEED INTO THE CONTROL      019236
C                                   SYSTEM (L=K-NBTQ)                       019237
C           FOR INDEX I;I=1,....,NY2+NXSS AND K;K=1,....,NBTQ               019238
C              B(I,K) = -A(I,K)   = (I,K) ELEMENT OF THE REFERENCE INPUT    019239
C                                   COEFFICIENT MATRIX, THESE ELEMENTS ARE  019240
C                                   ASSOCIATED WITH REFERENCE INPUT TORQUE  019241
C                                   SIGNALS FEED INTO THE PLANT             019242
C           FOR INDEX I;I=1,....,NY2+NXSS AND K;K=NBTQ+1,....,NBTQ+NXSS     019243
C              B(I,K) = 0                                                   019244
C           FOR INDEX I;I=NY2+NXSS+1,....,NX AND K;K=1,....,NBTQ            019245
C              B(I,K) = 0                                                   019246
C           FOR INDEX I;I=NY2+NXSS+1,....,NX AND K;K=NBTQ+1,....,NBTQ+NXSS  019247
C              B(I,K) = A(I,K)    = (I,K) ELEMENT OF THE REFERENCE INPUT    019248
C                                   COEFFICIENT MATRIX, THESE ELEMENTS ARE  019249
C                                   ASSOCIATED WITH REFERENCE INPUT SENSOR  019250
C                                   SIGNALS FEED INTO THE CONTROL           019251
C                                                                           019252
C        IN GENERAL THE ELEMENTS X(I,S),I=1,....,NX DEFINE ALL POSSIBLE     019253
C           OUTPUT SIGNALS AND THE ELEMENTS U(K,S),K=1,....,NAUX DEFINE     019254
C           ALL POSSIBLE INPUT SENSOR AND TORQUE REFERENCE SIGNALS. IN      019255
C           THE S-DOMAIN THE EQUATIONS MAY BE EXPRESSED AS                  019256
C                                                                           019257
C                (I(I,J)*S - A(I,J))*X(J,S) = B(I,K)*U(K,S)                 019258
C                                                                           019259
C           WHERE                                                           019260
C                    I(I,J) = 1  IF(I.EQ.J)                                 019261
C                    I(I,J) = 0  IF(I.NE.J)                                 019262
C                    I = 1,....,NX                                          019263
C                    J = 1,....,NX                                          019264
C                    K = 1,....,NAUX                                        019265
```

```
C                                                                        019266
C                                                                        019267
C     THE COEFFICIENT MATRIX 'A' COMPLETELY DEFINES ALL COUPLING BETWEEN 019268
C         THE NX LINEARLY INDEPENDENT STATE VARIABLES, SENSOR SIGNALS    019269
C         AND TORQUE SIGANLS USED TO MODEL THE DYNAMIC RESPONSE OF THE   019270
C         SYSTEM                                                         019271
C                                                                        019272
C     SEVERAL DIFFERENT TYPES OF TRANSFER FUNCTIONS ARE USED IN THE      019273
C         LINEAR STABILITY ANALYSIS OF A PARTICULAR SYSTEM. THE 'TYPES   019274
C         OF TRANSFER FUNCTIONS' MAY BE CLASSIFIED ACCORDING TO WHAT     019275
C         FEEDBACK INFORMATION IS ALLOWED; THAT IS, WHICH FEEDBACK LOOPS 019276
C         ARE OPEN AND WHICH FEEDBACK LOOPS ARE CLOSED.                  019277
C                                                                        019278
C     FEEDBACK LOOPS ARE CUT IN THE DISCOS FORMULATION BY SELECTIVELY    019279
C         SETTING CERTAIN ELEMENTS IN THE 'A' MATRIX EQUAL TO ZERO TO    019280
C         FORM THE REDUCED MATRIX 'AR'. THE ASSOCIATED TRANSFER FUNCTION 019281
C         WHICH DEFINES THE OUTPUT RESPONSE OF THE P-TH STATE VARIABLE   019282
C         X(P,S) TO THE Q-TH REFERENCE INPUT SIGNAL U(Q,S) IS FROM       019283
C         CRAMER'S RULE DEFINED BY                                       019284
C                                                                        019285
C                   X(P,S)    DET(AUG(I*S - AR))                         019286
C                   ------ = ------------------                          019287
C                   U(Q,S)       DET(I*S - AR)                           019288
C                                                                        019289
C         WHERE                                                          019290
C                       I = UNIT DIAGONAL MATRIX                         019291
C          AUG(I*S - AR) = AUGMENTED MATRIX OBTAINED BY PLACING COLUMN   019292
C                          Q OF THE 'B' MATRIX (REFERENCE INPUT SIGNAL   019293
C                          COEFFICIENT MATRIX)INTO COLUMN P OF THE       019294
C                          '(I*S - AR)' MATRIX                           019295
C                    DET = 'DETERMINANT OF'                              019296
C                                                                        019297
C     SUBROUTINE TFTYPE ACCEPTS FROM DYNS40 THE COEFFICIENT MATRIX 'A'   019298
C         AND INTEGER CODES WHICH DEFINE WHICH FEEDBACK LOOPS ARE TO BE  019299
C         CUT AND WITH RESPECT TO WHICH REFERENCE INPUT SIGNAL THE       019300
C         TRANSFER FUNCTION IS TO BE OBTAINED. IT RETURNS TO DYNS40      019301
C         THE REDUCED MATRIX 'AR', ITS DIMENSION AND THE COLUMN OF THE   019302
C         'B' MATRIX ASSOCIATED WITH THE REFERENCE INPUT SIGNAL OF       019303
C         INTEREST                                                       019304
C                                                                        019305
C     ITYPE  - INTEGER CODE USED TO DEFINE THE 'TYPE OF TRANSFER         019306
C              FUNCTION' TO BE GENERATED                                 019307
C     JCOL   = INTEGER CODE USED TO DEFINE PARTICULAR REFERENCE INPUT    019308
C              SIGNAL OF INTEREST                                        019309
C     A      = DUMMY ARRAY IN WHICH THE 'A' MATRIX IS PASSED TO TFTYPE   019310
C     Z      = DUMMY ARRAY IN WHICH THE 'AR' MATRIX IS RETURNED          019311
C     E      = DUMMY ARRAY IN WHICH THE COLUMN OF THE 'B' MATRIX         019312
C              ASSOCIATED WITH THE REFERENCE INPUT SIGNAL OF INTEREST    019313
C              IS RETURNED                                               019314
C     NZ     = INTEGER, DIMENSION OF THE 'AR' MATRIX                     019315
C     NGKP   = SPECIAL FOR ITYPE=7                                       019316
C     KIKP   = SPECIAL FOR ITYPE=7                                       019317
C                                                                        019318
C                                                                        019319
C                -- TRANSFER FUNCTION SETUP --                           019320
C                                                                        019321
C         ITYPE = 1 = FORWARD PATH TRANSFER FUNCTION G(S). THE           019322
C                     REDUCED MATRIX 'AR' IS (NY2+NXSS) BY (NY2+NXSS)    019323
C                     THE REFERENCE INPUT TORQUE SIGNAL OF INTEREST IS   019324
C                     TORQUE SIGNAL JCOL WHERE  0 < JCOL < NBTQ+1.       019325
```

```
C                     THE FOLLOWING CODE IS USED IN TFTYPE              019326
C                          NZ = NY2+NXSS                                019327
C                          JB = NY2+NXSS+ND2+JCOL                       019328
C                          DO 1 I=1,NZ                                  C19329
C                          B(I) = A(I,JB)                               019330
C                          DO 1 J=1,NZ                                  019331
C                          Z(I,J) = A(I,J)                              019332
C                        1 CONTINUE                                     019333
C                     IF(ITYPE.EQ.-1) THE SIGNS OF THE ELEMENTS IN 'B'  C19334
C                     ARE ALL CHANGED (NEGATIVE FEEDBACK FOR NUMERATOR) 019335
C                     THE TRANSFER FUNCTION BETWEEN ANY OUTPUT SENSOR   019336
C                     SIGNAL AND REFERENCE INPUT TORQUE SIGNAL JCOL     C19337
C                     MAY BE OBTAINED                                   C19338
C                                                                       C19339
C         ITYPE =  2 = FEEDBACK PATH TRANSFER FUNCTION H(S). THE        019340
C                     REDUCED MATRIX 'AR' IS (ND2+NBTQ) BY (ND2+NBTQ)   019341
C                     THE REFERENCE INPUT SENSOR SIGNAL OF INTEREST IS  019342
C                     SENSOR SIGNAL JCOL WHERE  0 < JCOL < NXSS+1       019343
C                     THE FOLLOWING CODE IS USED IN TFTYPE              019344
C                          NZ = ND2+NBTQ                                019345
C                          JB = NY2+JCOL                                C19346
C                          DO 1 I=1,NZ                                  C19347
C                          II = NY2+NXSS+I                              C19348
C                          B(I) = A(II,JB)                              019349
C                          DO 1 J=1,NZ                                  C19350
C                          JJ = NY2+NXSS+J                              019351
C                          Z(I,J) = A(II,JJ)                            C19352
C                        1 CONTINUE                                     019353
C                     THE TRANSFER FUNCTION BETWEEN ANY OUTPUT TORQUE   C19354
C                     SIGNAL AND REFERENCE INPUT SENSOR SIGNAL JCOL     019355
C                     MAY BE OBTAINED                                   019356
C                                                                       C19357
C         ITYPE =  3 = OPEN LOOP TRANSFER FUNCTION G(S)*H(S). THE       019358
C                     FEEDBACK PATH FROM THE CONTROL SYSTEM TO THE      019359
C                     PLANT IS COMPLETELY CUT. THIS IS EFFECTED BY      C19360
C                     SETTING ALL COEFFICIENTS ASSOCIATED WITH CONTROL  019361
C                     SYSTEM STATE VARIABLES, IN THE EQUATIONS FOR THE  019362
C                     PLANT STATE VARIABLES, EQUAL TO ZERO. THE REDUCED 019363
C                     MATRIX 'AR' IS NX BY NX AND THE REFERENCE INPUT   C19364
C                     TORQUE SIGNAL OF INTEREST IS TORQUE SIGNAL JCOL   C19365
C                     WHERE  0 < JCOL < NBTQ+1. THE FOLLOWING CODE IS   C19366
C                     USED IN TFTYPE                                    C19367
C                          NZ = NX                                      019368
C                          NN = NY2+NXSS                                C19369
C                          JB = NY2+NXSS+ND2+JCOL                       019370
C                          DO 1 I=1,NX                                  019371
C                          B(I) = 0.0D0                                 C19372
C                          IF(I.LE.NN) B(I) = A(I,JB)                   019373
C                          DO 1 J=1,NX                                  019374
C                          Z(I,J) = 0.0D0                               C19375
C                          IF(I.LE.NN.AND.J.GT.NN) GO TO 1              019376
C                          Z(I,J) = A(I,J)                              C19377
C                        1 CONTINUE                                     019378
C                     IF(ITYPE.EQ.-3) THE SIGNS OF THE ELEMENTS IN 'B'  019379
C                     ARE ALL CHANGED. THE TRANSFER FUNCTION BETWEEN    C19380
C                     ANY OUTPUT TORQUE SIGNAL AND REFERENCE INPUT      C19381
C                     TORQUE SIGNAL JCOL MAY BE OBTAINED                019382
C                                                                       C19383
C         ITYPE =  4 = OPEN LOOP TRANSFER FUNCTION H(S)*G(S). THE       C19384
C                     FEEDBACK PATH FROM THE PLANT TO THE CONTROL       C19385
```

```
C                 SYSTEM IS COMPLETELY CUT. THIS IS EFFECTED BY         019386
C                 SETTING ALL COEFFICIENTS ASSOCIATED WITH PLANT        C19387
C                 STATE VARIABLES, IN THE EQUATIONS FOR THE CONTROL     C19388
C                 STATE VARIABLES, EQUAL TO ZERO. THE REDUCED           C19389
C                 MATRIX 'AR' IS NX BY NX AND THE REFERENCE INPUT       019390
C                 SENSOR SIGNAL OF INTEREST IS JCOL WHERE               C19391
C                 0 < JCOL < NXSS+1. THE FOLLOWING CODE IS USED         019392
C                 IN TFTYPE                                             019393
C                     NZ = NX                                           C19394
C                     NN = NY2+NXSS                                     C19395
C                     JB = NY2+JCOL                                     019396
C                     DO 1 I=1,NX                                       C19397
C                     B(I) = 0.0D0                                      C19398
C                     IF(I.GT.NN) B(I) = A(I,JB)                        C19399
C                     DO 1 J=1,NX                                       C19400
C                     Z(I,J) = 0.0D0                                    C19401
C                     IF(I.GT.NN.AND.J.LE.NN) GO TO 1                   C19402
C                     Z(I,J) = A(I,J)                                   019403
C                   1 CONTINUE                                          019404
C                 THE TRANSFER FUNCTION BETWEEN ANY OUTPUT SENSOR       C19405
C                 SIGNAL AND REFERENCE INPUT SENSOR SIGNAL JCOL         019406
C                 MAY BE OBTAINED                                       C19407
C                                                                       C19408
C         ITYPE =  5 - CLOSED LOOP TRANSFER FUNCTION - CONTROL RATIO    C19409
C                 G(S)/(1+G(S)*H(S)). ALL FEEDBACK PATHS ARE CLOSED     019410
C                 AND HENCE THE REDUCED MATRIX 'AR' IS NX BY NX AND     C19411
C                 EQUAL TO THE INPUTTED 'A' MATRIX. THE REFERENCE       019412
C                 INPUT TORQUE SIGNAL OF INTEREST IS JCOL WHERE         019413
C                 0 < JCOL < NITC+1. THE FOLLOWING CODE IS USED         C19414
C                 IN TFTYPE                                             019415
C                     NZ = NX                                           C19416
C                     NN = NY2+NXSS                                     C19417
C                     JB = NY2+NXSS+ND2+JCOL                            019418
C                     DO 1 I=1,NX                                       019419
C                     B(I) = 0.0D0                                      019420
C                     IF(I.LE.NN) B(I) = A(I,JB)                        019421
C                     DO 1 J=1,NX                                       019422
C                     Z(I,J) - A(I,J)                                   019423
C                   1 CONTINUE                                          019424
C                 IF(ITYPE.EQ.-5) THE SIGNS OF THE ELEMENTS IN 'B'      C19425
C                 ARE ALL CHANGED. THE TRANSFER FUNCTION BETWEEN        019426
C                 ANY OUTPUT SENSOR SIGNAL AND THE REFERENCE INPUT      019427
C                 TORQUE SIGNAL JCOL MAY BE OBTAINED                    019428
C                                                                       C19429
C         ITYPE =  6 = CLOSED LOOP TRANSFER FUNCTION                    019430
C                 G(S)*H(S)/(1+G(S)*H(S)). ALL FEEDBACK PATHS ARE       C19431
C                 CLOSED AND HENCE THE REDUCED MATRIX 'AR' IS           019432
C                 NX BY NX AND EQUAL TO THE INPUTTED 'A' MATRIX.        C19433
C                 THE REFERENCE INPUT SENSOR SIGNAL OF INTEREST IS      019434
C                 JCOL WHERE 0 < JCOL < NXSS+1. THE FOLLOWING CODE      C19435
C                 IS USED IN TFTYPE                                     C19436
C                     NZ = NX                                           C19437
C                     NN = NY2+NXSS                                     019438
C                     JB = NY2+JCOL                                     019439
C                     DO 1 I=1,NX                                       019440
C                     B(I) = 0.0D0                                      C19441
C                     IF(I.GT.NN) B(I) = A(I,JB)                        C19442
C                     DO 1 J=1,NX                                       C19443
C                     Z(I,J) = A(I,J)                                   019444
C                   1 CONTINUE                                          019445
```

```
C                     THE TRANSFER FUNCTION BETWEEN ANY OUTPUT SENSOR    C19446
C                     SIGNAL AND THE REFERENCE INPUT SENSOR SIGNAL       C19447
C                     JCOL MAY BE OBTAINED                               C19448
C                                                                        C19449
C         ITYPE =  -7 - QUASI OPEN LOOP TRANSFER FUNCTION. THE FEEDBACK  C19450
C                     PATH FROM THE CONTROL SYSTEM TO THE PLANT IS       C19451
C                     PARTIALLY CLOSED. THE NBKP TORQUE SIGNALS          C19452
C                     KBKP(1).....KBKP(NBKP) ARE THE ONLY TORQUE         C19453
C                     SIGNALS FEEDBACK FROM THE CONTROL SYSTEM TO THE    C19454
C                     PLANT. ALL OTHER LOOPS ARE OPEN (NBKP.LE.3). THIS  C19455
C                     IS EFFECTED BY SETTING ALL COEFFICIENTS            C19456
C                     ASSOCIATED WITH THE CONTROL SYSTEM STATE           C19457
C                     VARIABLES, IN THE EQUATIONS FOR THE PLANT STATE    C19458
C                     VARIABLES, EQUAL TO ZERO EXCEPT THOSE ASSOCIATED   C19459
C                     WITH THE NBKP TORQUE SIGNALS TO BE FEEDBACK. THE   C19460
C                     REDUCED MATRIX 'AR' IS NA BY NA AND THE REFERENCE  C19461
C                     INPUT TORQUE SIGNAL OF INTEREST IS TORQUE SIGNAL   C19462
C                     JCOL WHERE 0 < JCOL < NKSS+1. THE FOLLOWING CODE   C19463
C                     IS USED IN TFTYPE                                  C19464
C                          NZ = NA                                       C19465
C                          NN = NY2+NKSS                                 C19466
C                          JB = NY2+NKSS+NJ2+JCOL                        C19467
C                          DO 1 I=1,NA                                   C19468
C                          C(I) = 0.0D0                                  C19469
C                          IF(I.LE.NN) C(I) = A(I,JB)                    C19470
C                          DO 1 J=1,NA                                   C19471
C                          Z(I,J) = A(I,J)                               C19472
C                          IF(J.LE.NN.OR.I.GT.NN) GO TO 1                C19473
C                          Z(I,J) = 0.0D0                                C19474
C                          DO 2 K=1,NBKP                                 C19475
C                          LB = NY2+NKSS+NJ2+KBKP(K)                     C19476
C                          IF(J.EQ.LB) Z(I,J) = A(I,LB)                  C19477
C                        2 CONTINUE                                      C19478
C                        1 CONTINUE                                      C19479
C                     IF(ITYPE.EQ.-7) THE SIGNS OF THE ELEMENTS IN 'B'   C19480
C                     ARE ALL CHANGED. THE TRANSFER FUNCTION BETWEEN     C19481
C                     ANY OUTPUT TORQUE SIGNAL AND THE REFERENCE INPUT   C19482
C                     TORQUE SIGNAL JCOL MAY BE OBTAINED                 C19483
C                                                                        C19484
C                       *****  SPECIAL CASE  *****                       C19485
C                                                                        C19486
C     A REFERENCE INPUT SIGNAL ASSOCIATED WITH ANY SYSTEM STATE          C19487
C         VARIABLE X(I,3),I=1,.......NX MAY BE FEED INTO THE SYSTEM.     C19488
C         EITHER ALL STATE VARIABLES OR ALL EXCEPT THE ONE ASSOCIATED    C19489
C         WITH THE REFERENCE INPUT SIGNAL MAY BE FEEDBACK TO THE SYSTEM. C19490
C         EITHER THE CLOSED LOOP OR THE ONE STATE OPEN LOOP TRANSFER     C19491
C         FUNCTION MAY BE OBTAINED BETWEEN ANY STATE VARIABLE AND THE    C19492
C         REFERENCE INPUT SIGNAL                                         C19493
C                                                                        C19494
C         ITYPE =  -8 - ONE STATE OPEN LOOP TRANSFER FUNCTION.           C19495
C                     A REFERENCE INPUT SIGNAL MAY BE SUMMED WITH ANY    C19496
C                     SYSTEM STATE VARIABLE X(JCOL,3), 0 < JCOL < NX+1   C19497
C                     AND FEED INTO THE TOTAL SYSTEM. FOR THIS CASE      C19498
C                     THE LOOP WHICH DEFINES X(JCOL,3) FEEDBACK INTO     C19499
C                     THE SYSTEM IS CUT. THIS IS EFFECTED BY SETTING     C19500
C                     TO ZERO ALL STATE VARIABLE COUPLING COEFFICIENTS   C19501
C                     ASSOCIATED WITH STATE VARIABLE JCOL FEEDBACK. THE  C19502
C                     FOLLOWING CODE IS USED IN TFTYPE                   C19503
C                          NZ = NA                                       C19504
C                          JB = JCOL                                     C19505
```

```
C                          DO 1 I=1,NX
C                          B(I) = 0.0D0
C                          IF(I.NE.JB) B(I) = -A(I,JB)
C                          DO 1 J=1,NX
C                          Z(I,J) = A(I,J)
C                          IF(J.NE.JB) GO TO 1
C                          IF(I.EQ.JB) GO TO 1
C                          Z(I,J) = 0.0D0
C                        1 CONTINUE
C                      THE TRANSFER FUNCITON BETWEEN ANY STATE VARIABLE
C                      AND ANY REFERENCE INPUT SIGNAL MAY BE OBTAINED
C                      (NEGATIVE FEEDBACK USED)
C
C         ITYPE = -6 = CLOSED LOOP TRANSFER FUNCTION. A REFERENCE INPUT
C                      SIGNAL MAY BE SUMMED WITH ANY SYSTEM STATE
C                      VARIABLE X(JCOL,S), 0 < JCOL < NX+1 AND FEED INTO
C                      THE TOTAL SYSTEM. FOR THIS CASE ALL FEEDBACK
C                      LOOPS ARE CLOSED. THE FOLLOWING CODE IS USED
C                      IN TFTYPE
C                          NZ = NX
C                          JB = JCOL
C                          DO 1 I=1,NX
C                          B(I) = 0.0D0
C                          IF(I.NE.JB) B(I) = -A(I,JCOL)
C                          DO 1 J=1,NX
C                          Z(I,J) = A(I,J)
C                        1 CONTINUE
C                      THE TRANSFER FUNCTION BETWEEN ANY STATE VARIABLE
C                      AND ANY REFERENCE INPUT SIGNAL MAY BE OBTAINED
C                      (NEGATIVE FEEDBACK USED)
C
C     IF DEBUG(67) = .TRUE. THE FOLLOWING CAPTIONS AND MATRICES ARE
C          PRINTED:
C     ' SUBROUTINE TFTYPE, TRANSFER = INPUT TRANSFER FUNCTION INTEGER
C       FUNCTION TYPE'                 CODE ITYPE
C     ' THE CHARACTERISTIC MATRIX   = THE NZ BY NZ REDUCED MATRIX 'AR'
C        IS'                           WHICH IS TO BE RETURNED TO DYNS40
C                                      IN THE DUMMY ARRAY Z
C     ' THE CORRESPONDING INPUT     = THE NZ BY 1 COLUMN MATRIX 'B'
C       COEFFICIENTS ARE'              WHICH IS TO BE RETURNED TO DYNS40
C                                      IN THE DUMMY ARRAY B
C
C
C         THE TRANSFER FUNCTION IN FACTORED POLYNOMIAL FORM
C         ---------------------------------------------
C
C     SUBROUTINE TFTYPE ACCEPTS THE COEFFICIENT MATRIX 'A' OF THE
C         LINEARIZED EQUATIONS OF MOTION. IT ALSO ACCEPTS THE USER
C         DEFINED INTEGER CODES WHICH DEFINE WHAT TYPE OF TRANSFER
C         FUNCTION IS DESIRED AND WHICH STATE VARIABLE WILL HE HAVE ITS
C         FEEDBACK MODIFIED BY A REFERENCE INPUT SIGNAL.
C
C     SUBROUTINE TFTYPE RETURNS TO DYNS40 THE REDUCED MATRIX 'AR' WHICH
C         IS THE CHARACTERISTIC MATRIX FOR THE TRANSFER FUNCITON OF
C         INTEREST. IT ALSO RETURNS THE COLUMN MATRIX 'B' WHICH IS THE
C         REFERENCE INPUT SIGNAL COEFFICIENT MATRIX, AND NZ ITS ORDER
C
C     UPON RETURNING TO DYNS40 FROM TFTYPE THE FOLLOWING CAPTIONS AND
C         MATRICES WILL BE PRINTED:
C     ' OUTPUT MATRIX 'AR' (NZ X NZ) = CHARACTERISTIC MATRIX COMPUTED
```

```
C                                     IN TFTYPE AND RETURNED TO DYNS40
C                                     IN DUMMY ARRAY W1. THIS IS THE
C                                     REDUCED MATRIX 'AR'
C     * OUTPUT MATRIX BCOL ( 1 X NZ) = THE NZ BY 1 COLUMN MATRIX 'B'
C                                     COMPUTED IN TFTYPE AND RETURNED
C                                     TO DYNS40 IN DUMMY ARRAY V1. THIS
C                                     IS THE COLUMN MATRIX OF REFERENCE
C                                     INPUT SIGNAL COEFFICIENTS TO BE
C                                     USED FOR TRANSFER FUNCITON
C                                     COMPUTATION
C
C     THESE MATRICES ALONG WITH OTHER STORED QUANTITIES PROVIDE
C         SUFFICIENT DATA TO CONSTRUCT THE DESIRED TRANSFER FUNCTION VIA
C         APPLICATION OF CRAMER'S RULE. THAT IS
C
C                  X(P.S)   DET(ADJ(I*S - AR))
C                  ------ = ------------------
C                  U(Q.S)     DET(I*S - AR)
C
C         WHERE THE DESIRED OUTPUT SIGNAL X(P.S) IS DETERMINED FROM THE
C         INTEGER CODES ITFOUT AND ITYPE.
C
C
C         DENOMINATOR ROOTS
C         -----------------
C
C     THE FACTORED POLYNOMIAL FORM OF THE DENOMINATOR IS FOUND BY A
C         SIMPLE CALL TO THE QR-ALGORITHM WITH THE MATRIX 'AR' TO FIND
C         ITS COMPLEX ROOTS. THE USER MAY REQUEST THAT 'AR TRANSPOSE'
C         BE USED INSTEAD OF 'AR', THIS AT TIMES FOR SPECIAL PROBLEMS
C         CAN BE USEFUL IN OVERCOMING NUMERICAL PROBLEMS. IN EITHER CASE
C         THE ROOTS OF 'AR' AND 'AR**T' ARE BOTH FOUND AND PRINTED UNDER
C         THE FOLLOWING CAPTIONS:
C          'R AR'                      = A LISTING IN DECREASING ORDER
C      'NO  REAL PART IMAGINARY PART'    OF REAL PART OF ALL NZ COMPLEX
C                                        ROOTS OF THE 'AR' MATRIX
C          'RART'                      = A LISTING IN DECREASING ORDER
C      'NO  REAL PART IMAGINARY PART'    OF REAL PART OF ALL NZ COMPLEX
C                                        ROOTS OF THE 'AR**T'MATRIX
C            ** NOTE **
C       BOTH SETS OF COMPLEX ROOTS
C       SHOULD BE EQUAL. THIS IS
C       A CHECK, THE ROOTS OF 'AR'
C       WILL BE USED BY DEFAULT
C
C     LET:
C          RRO(K) = REAL PART OF K-TH COMPLEX ROOT OF 'AR'
C          RIO(K) = IMAGINARY PART OF K-TH COMPLEX ROOT OF 'AR'
C     THEN
C          DET(I*S - AR) =  (S - RRO( 1) - J*RIO( 1))
C                          *(S - RRO( 2) - J*RIO( 2))
C                                       .
C                                       .
C                                       .
C                                       .
C                          *(S - RRO(NZ) - J*RIO(NZ))
C     WHERE
C           J=SQRT(-1.0)
C     AND REAL AND IMAGINARY PARTS .LE. TOLD ARE SET TO ZERO
C
```

```
C                                                                       019626
C        NUMERATOR ROOTS                                                019627
C        ---------------                                                019628
C                                                                       019629
C     THE FACTORED POLYNOMIAL FORM OF THE NUMERATOR IS NOT AS STRAIGHT- 019630
C        FORWARD TO OBTAIN. THIS IS DONE IN THE SUBROUTINE NUMS         019631
C        DISCUSSED BELOW. THE DIFFICULTY IN OBTAINING THE ROOTS OF      019632
C        THE NUMERATOR COMES FROM THE FACT THAT ITS DEGREE MAY (AND     019633
C        USUALLY IS) LESS THAN THAT OF THE DENOMINATOR.                 019634
C                                                                       019635
C     NUMS RECEIVES FROM DYNS4D THE REDUCED MATRIX 'AR', THE COLUMN     019636
C        MATRIX 'B' AND THE COLUMN LOCATION P OF THE OUTPUT STATE       019637
C        VARIABLE X(P,J). IT FORMS THE AUGMENTED MATRIX 'AUG(I*S - AR)' 019638
C        BY REPLACING COLUMN P OF '(I*S - AR)' WITH THE 'B' COLUMN      019639
C        MATRIX TO FORM  'AUG(I*S - AR)'. USING SPECIAL TECHNIQUES      019640
C        THE DETERMINATE IS FORMED AND RETURNED TO DYNS4D IN FACTORED   019641
C        POLYNOMIAL FORM WHERE THE FOLLOWING CAPTIONS AND MATRICES      019642
C        ARE PRINTED:                                                   019643
C         ' NUM '                     = A LISTING IN DECREASING ORDER   019644
C      'NO  REAL PART IMAGINARY PART'     OF REAL PARTS OF ALL COMPLEX  019645
C                                         ROOTS OF THE TRANSFER FUNCTION019646
C                                         NUMERATOR.                    019647
C         ' DEN'                      = A LISTING IN DECREASING ORDER   019648
C      'NO  REAL PART IMAGINARY PART'     OF REAL PARTS OF ALL COMPLEX  019649
C                                         ROOTS OF THE TRANSFER FUNCTION019650
C                                         DENOMINATOR                   019651
C            ** NOTE **                                                 019652
C       IF THE NUMBER OF NUMERATOR                                      019653
C       ROOTS NN IS LESS THAN THE                                       019654
C       NUMBER OF DENOMINATOR ROOTS                                     019655
C       NZ THEN UNDER THE TWO COLUMNS                                   019656
C       WITH CAPTION HEADING 'NUM'                                      019657
C       THE LAST (NZ-NN) ROWS WILL                                      019658
C       BE BLANK                                                        019659
C                                                                       019660
C     LET:                                                              019661
C         GAIN   = NUMERATOR ROOT GAIN                                  019662
C         NN     = TOTAL NUMBER OF NUMERATOR ROOTS                      019663
C         RRN(K) = REAL PART OF K-TH COMPLEX ROOT OF NUMERATOR          019664
C         RIN(K) = IMAGINARY PART OF K-TH COMPLEX ROOT OF NUMERATOR     019665
C     THEN                                                              019666
C        DET(AUG(I*S - AR)) =  GAIN*(S - RRN( 1) - J*RIN( 1))           019667
C                                  *(S - RRN( 2) - J*RIN( 2))           019668
C                                               .                       019669
C                                               .                       019670
C                                               .                       019671
C                                  *(S - RRN(NN) - J*RIN(NN))           019672
C     WHERE                                                             019673
C          J=SQRT(-1.0)                                                 019674
C     AND REAL AND IMAGINARY PARTS .LE. TOLN ARE SET TO ZERO            019675
C                                                                       019676
C                                                                       019677
C        NUMS - FIND TRANSFER FUNCTION NUMERATOR ROOTS                  019678
C        ---------------------------------------------                  019679
C                                                                       019680
C     THE COMPUTATION OF THE ROOTS OF THE NUMERATOR PRESENTS A PROBLEM  019681
C        WHICH MUST BE OVERCOME. THE ALGORITM MUST BE ABLE TO DETERMINE 019682
C        NOT ONLY THE DEGREE OF THE POLYNOMIAL BUT ALSO ITS ROOTS.      019683
C        THE METHOD USED MAKES A QUICK CHECK TO SEE IF THE PIVOT        019684
C        ELEMENT IS ZERO; IF NOT, BOTH NUMERATOR AND DENOMINATOR HAVE   019685
```

```
C         THE SAME DEGREE AND NUMERATOR ROOTS ARE READILY OBTAINABLE      019686
C         BY A DIRECT APPLICATION OF THE QR-ALGORITHM.                    019687
C         IF THE PIVOT ELEMENT IS ZERO THE DEGREE OF THE NUMERATOR IS     019688
C         AT LEAST 1 LESS THAN THE DENOMINATOR. THE THEORY USED TO SOLVE  019689
C         THIS PROBLEM IS ON PAGES 71-73, VOL I.                          019690
C         BASICALLY, THE SUBSTITUTION                                     019691
C                                                                         019692
C                         (S - CON) = 1.0/P                               019693
C                                                                         019694
C         IS MADE IN THE NUMERATOR WHERE CON IS AN ARBITRARY CONSTANT     019695
C         NOT EQUAL TO A ROOT OF THE NUMERATOR. IT HAS BEEN ARBITRARILLY  019696
C         SET AS                                                          019697
C                           CON = -DSQRT(3.D0)                            019698
C         IN DISCUS                                                       019699
C                                                                         019700
C         THIS EVENTUALLY LEADS TO THE NEED TO DETERMINE THE ROOTS OF A   019701
C         DETERMINANT EXPRESSION WHICH IS OF THE FORM                     019702
C                                                                         019703
C                          DET(I*P - D)                                   019704
C                                                                         019705
C         WHERE D IS A DERIVABLE NZ BY NZ MATRIX, SEE EQ III-41 VOL I.    019706
C         ALL NULL ROOTS IMPLY ROOTS AT INFINITY OR A CHARACTERISTIC      019707
C         POLYNOMIAL HAVING ORDER LESS THAN NZ. THUS:                     019708
C                                                                         019709
C                    NN = ORDER OF NUMERATOR                              019710
C                       = NZ - NUMBER OF ZERO ROOTS OF DET(I*P - D)       019711
C         AND                                                             019712
C                S(I) = I-TH ROOT OF NUMERATOR IS OBTAINED FROM           019713
C                                                                         019714
C                     S(I) = CON + 1.0/P(I)     (COMPLEX ARITHMETIC)      019715
C                                                                         019716
C         WHERE                                                           019717
C                P(I) = I-TH NON-ZERO ROOT OF DET(I*P - D)                019718
C                                                                         019719
C     THE FOLLOWING PARAMETERS ARE PASSED TO NUMS FROM DYNS40 AND         019720
C         RETURNED. BY SETTING DEBUG(68) = .TRUE. SEVERAL MAY BE PRINTED  019721
C         ALONG WITH THE DESCRIPTIVE CAPTIONS INDICATED.                  019722
C     A       = INPUTTED NZ BY NZ REDUCED MATRIX 'AR' WHICH WAS SETUP IN  019723
C                 TFTYPE. ARRAY DESTROYED DURING COMPUTATION              019724
C     C       = DUMMY ARRAY USED AS WORK SPACE                            019725
C     B       = COLUMN OF THE 'B' MATRIX SETUP IN TFTYPE                  019726
C     RIR     = OUTPUT OF NUMS. THE NZRO REAL PARTS OF THE COMPLEX ROOTS  019727
C                 OF THE NUMERATOR                                        019728
C     RII     = OUTPUT OF NUMS. THE NZRO IMAGINARY PARTS OF THE COMPLEX   019729
C                 ROOTS OF THE NUMERATOR                                  019730
C     R2R     = WORK AREA                                                 019731
C     R2I     = WORK AREA                                                 019732
C     PTOL    = TOLERANCE FACTOR ALL P-ROOTS LESS THAN PTOL SET TO ZERO   019733
C     GAIN    = OUTPUT. NUMERATOR GAIN SEE EQ III-48 VOL I                019734
C     IFLG    = 0 IF ERROR ZERO NUMERATOR GAIN, =1 IF NO ERROR            019735
C     NN      = 1                                                         019736
C     NZRO    = OUTPUT THE ORDER OF AND HENCE NUMBER OF NUMERATOR ROOTS   019737
C     IY      = COLUMN LOCATION IN 'AR' OF DESIRED OUTPUT SIGNAL FOR      019738
C                 TRANSFER FUNCTION                                       019739
C     NA      = DIMENSION OF 'AR' (DEFINED AS NZ ABOVE)                   019740
C                                                                         019741
C      * ANUM  (NA X NA) *      = PRINTED IF THE ORDER OF THE NUMERATOR   019742
C                                 EQUAL TO THAT OF THE DENOMINATOR. IT    019743
C                                 IS THE (NZ BY NZ) AUGMENTED MATRIX      019744
C                                 PASSED TO THE QR-ALGORITHM (NA=NZ)      019745
```

```
C       ' -C-    (NA X NA) '      = PRINTED IF THE ORDER OF THE NUMERATOR    019746
C                                   LESS THAN THAT OF THE DENOMINATOR. IT    019747
C                                   IS THE MATRIX 'D' DEFINED ABOVE USED     019748
C                                   TO OBTAIN THE ROOTS OF DET(I*P - D)      019749
C       '-RIR-  (1 X NZRO)'       = THE REAL PARTS OF THE NZRO COMPLEX       019750
C                                   ROOTS OF THE NUMERATOR                   019751
C       '-RII-  (1 X NZRO)'       = THE IMAGINARY PARTS OF THE NZRO COMPLEX  019752
C                                   ROOTS OF THE NUMERATOR                   019753
C       'GRATIO (1 X NA)  '       = THE NA PIVOT ELEMENTS OBTAINED IN THE    019754
C                                   PROCESS OF COMPUTING THE 'D' MATRIX      019755
C      GRVEC                        IN DET(I*P - D). THESE ARE STORED IN     019756
C                                   ARRAY GRVEC DIMENSIONED (IGASMX) IN      019757
C                                   /GRATIO/                                 019758
C       ' GAIN  (1 X 1) '         = NUMERATOR ROOT GAIN                      019759
C                                                                            C19760
C                                                                            C19761
C          PUT IN STANDARD BODE FORM                                         019762
C          -------------------------                                         019763
C                                                                            019764
C                                                                            C19765
C      FOR FREQUENCY RESPONSE CALCULATIONS IT IS CONVIENENT TO CONVERT       019766
C          POLES, ZEROS AND ROOT GAIN TO STANDARD BODE FORM. THAT IS TO      019767
C          WRITE THE TRANSFER FUNCTION IN TERMS OF TIME CONSTANTS,           C19768
C          DAMPING ZETAS AND RESONANT FREQUENCIES. THE RESPONSE IS THEN      C19769
C          EASILY GENERATE BY THE STANDARD SUBSTITUTION                      C19770
C                                                                            C19771
C                            S = J*OMEGA                                     C19772
C                                                                            019773
C          AND EVALUATION OF THE TRANSFER FUNCTION EXPRESSION AT VARIOUS     019774
C          VALUES OF OMEGA                                                   019775
C                                                                            019776
C      SUBROUTINE DCQRRT IS USED TO CLASSIFY AND COUNT THE ROOTS OF THE      C19777
C          DENOMINATOR AND NUMERATOR, WHILE SUBROUTINE DFORMB USES THE       C19778
C          OUTPUT OF DCQRRT TO COMPUTE TIME CONSTANT, DAMPING ZETA AND       019779
C          RESONANT FREQUENCY ARRAYS                                         C19780
C                                                                            C19781
C      THE NZ COMPLEX ROOTS OF THE DENOMINATOR ARE STORED TEMPORARLY IN      019782
C          THE ARRAYS RRD AND RID. THESE ARE PASSED TO DCQRRT AND THE        C19783
C          FOLLOWING COUNTS ARE RETURNED AND STORED IN /CCOUNT/:             019784
C      NDR      = NUMBER OF REAL AND ZERO DENOMINATOR ROOTS                  C19785
C      NDZ      = NUMBER OF ZERO DENOMINATOR ROOTS                           019786
C      ICD      = NUMBER OF COMPLEX PAIRS IN DENOMINATOR ROOTS ARRAY         C19787
C          THE DUMMY ARRAYS RERT AND CMPR ARE ALSO LOADED IN DCQRRT.         C19788
C          THESE ARE IN TURN PASSED TO DFORMB WHERE DENOMINATOR TIME         C19789
C          CONSTANTS, DAMPING ZETAS AND RESONANT FREQUENCIES ARE COMPUTED    019790
C          AND RETURNED IN THE FOLLOWING ARRAYS WHICH ARE STORED             019791
C          IN /CBRRAY/:                                                      019792
C      FDRD     = TIME CONSTANT ARRAY FOR REAL AND ZERO DENOMINATOR ROOTS    019793
C                 DIMENSIONED (LI)- ELEMENTS DEFINED ACCORDING TO            019794
C                      FDRD(I) = 0.000 IF ELEMENT RERT(I).EQ.0               019795
C                      FDRD(I) = TIME CONSTANT ASSOCIATED WITH THE I-TH      C19796
C                                ELEMENT IN THE REAL DENOMINATOR ROOTS       019797
C                                ARRAY RERT (I=1,.....NDR)                   019798
C      FDDC     = DAMPING ZETA, RESONANT FREQUENCY ARRAY FOR DENOMINATOR     C19799
C                 ROOTS WHICH APPEAR AS COMPLEX PAIRS. DIMENSIONED (LI)      019800
C                      FDDC(2*L-1) = DAMPING ZETA ASSOCIATED WITH THE L-TH   019801
C                                    COMPLEX ROOT PAIR IN THE DENOMINATOR    019802
C                                    ROOTS ARRAY CMPR                        019803
C                      FDDC(2*L)   = RESONANT FREQUENCY ASSOCIATED WITH THE  019804
C                                    L-TH COMPLEX ROOT PAIR IN THE           C19805
```

```
C                                   DENOMINATOR ROOTS ARRAY CMPR          C19806
C                                   (L=1,2,....ICD)                       C19807
C                                                                         C19808
C     THE NN COMPLEX ROOTS OF THE NUMERATOR ARE STORED TEMPORARLY IN      C19809
C         THE ARRAYS RRN AND RIN. THESE ARE PASSED TO DCQRRT AND THE      C19810
C         FOLLOWING COUNTS ARE RETURNED AND STORED IN /ZCOUNT/:           C19811
C     NNR     = NUMBER OF REAL AND ZERO NUMERATOR ROOTS                   C19812
C     NNZ     = NUMBER OF ZERO NUMERATOR ROOTS                            C19813
C     ICN     = NUMBER OF COMPLEX PAIRS IN NUMERATOR ROOTS ARRAY          C19814
C         THE DUMMY ARRAYS RERT AND CMPR ARE ALSO LOADED IN DCQRRT.       C19815
C         THESE ARE IN TURN PASSED TO DFCRMB WHERE NUMERATOR TIME         C19816
C         CONSTANTS, DAMPING ZETAS AND RESONANT FREQUENCIES ARE COMPUTED  C19817
C         AND RETURNED IN THE FOLLOWING ARRAYS WHICH ARE STORED           C19818
C         IN /ZRARAY/:                                                    C19819
C     FBNR    = TIME CONSTANT ARRAY FOR REAL AND ZERO NUMERATOR ROOTS     C19820
C               DIMENSIONED (LI), ELEMENTS DEFINED ACCORDING TO           C19821
C                    FBNR(I) = 0.000 IF ELEMENT RERT(I).EQ.0              C19822
C                    FBNR(I) = TIME CONSTANT ASSOCIATED WITH THE I-TH     C19823
C                              ELEMENT IN THE REAL NUMERATOR ROOTS        C19824
C                              ARRAY RERT (I=1,2,....NNR)                 C19825
C     FBNC    = DAMPING ZETA, RESONANT FREQUENCY ARRAY FOR NUMERATOR      C19826
C               ROOTS WHICH APPEAR AS COMPLEX PAIRS, DIMENSIONED (LI)     C19827
C                    FBNC(2*L-1) = DAMPING ZETA ASSOCIATED WITH THE L-TH  C19828
C                                  COMPLEX ROOT PAIR IN THE NUMERATOR     C19829
C                                  ROOTS ARRAY CMPR                       C19830
C                    FBNC(2*L)   = RESONANT FREQUENCY ASSOCIATED WITH THE C19831
C                                  L-TH COMPLEX ROOT PAIR IN THE          C19832
C                                  NUMERATOR ROOTS ARRAY CMPR             C19833
C                                  (L=1,2,....ICN)                        C19834
C                                                                         C19835
C     BY SETTING DEBUG(60) = .TRUE. THE FOLLOWING CAPTIONS AND            C19836
C         ASSOCIATED ARRAYS WILL BE PRINTED BEFORE EXIT FROM DFCRMB       C19837
C     ' OUTPUT MATRIX  RERT  (1 X KR)' = REAL AND ZERO ROOTS ARRAY        C19838
C                                        PASSED TO DFCRMB                 C19839
C     ' OUTPUT MATRIX  CMPR  (1 X KK)' = COMPLEX ROOTS ARRAY PASSED       C19840
C                                        TO DFCRMB                        C19841
C     ' OUTPUT MATRIX  FBR   (1 X KR)' = TIME CONSTANT ARRAY RETURNED     C19842
C                                        BY DFCRMB                        C19843
C     ' OUTPUT MATRIX  FBC   (1 X KK)' = DAMPING ZETA AND RESONANT        C19844
C                                        FREQUENCY ARRAY RETURNED BY      C19845
C                                        DFCRMB                           C19846
C     ' OUTPUT MATRIX  ACCO  (1 X 1) ' = GAIN FOR NUMERATOR, 1.0 FOR      C19847
C                                        DENOMINATOR RETURNED BY DFCRMB   C19848
C     ' OUTPUT MATRIX  GD    (1 X 2) ' = PRODUCT OF ALL NON ZERO ROOTS    C19849
C                                        RETURNED BY DFCRMB               C19850
C                                                                         C19851
C                                                                         C19852
C         DATA SETUP FOR FREQUENCY RESPONSE AND ROOT LOCUS COMPUTATION    C19853
C---------------------------------------------------------------------    C19854
C                                                                         C19855
C     THE NUMERICAL ALGORITHMS USED TO COMPUTE THE TRANSFER FUNCTION'S    C19856
C         FREQUENCY RESPONSE AND ROOT LOCUS DATA REQUIRE THAT THE POLE    C19857
C         ZERO DATA OBTAINED BE PUT IN A PARTULAR FORMAT. THIS IS         C19858
C         ACCOMPLISHED BY SUBROUTINES TTFF, IN WHICH THE ROOTS ARRAY 'R'  C19859
C         IS SETUP, CANCOR, IN WHICH ROOTS COMMON TO THE NUMERATOR AND    C19860
C         DENOMINATOR ARE CANCELLED AND RTOP, IN WHICH THE POLYNOMIAL     C19861
C         COEFFICIENTS ARRAY 'RA' IS SETUP.                               C19862
C                                                                         C19863
C                                                                         C19864
C         TTFF - SETUP ROOTS ARRAY 'R'                                    C19865
```

```
C         ------------------------------                               C19866
C                                                                      019867
C    R       = ROOTS ARRAY 'R' DIMENSIONED (2*L1+7) AND STORED         C19868
C              IN ZEROOTZ. ALL NUMERATOR AND DENOMINATOR ROOT COUNTS,  C19869
C               REAL ROOTS, COMPLEX ROOTS AND BODE GAIN DATA ARE STORED C19870
C              IN THE ROOTS ARRAY ACCORDING TO THE FOLLOWING FORMAT    C19871
C              R(1) = NUMBER OF REAL NON-ZERO NUMERATOR ROOTS          C19872
C              R(2) = NUMBER OF COMPLEX PAIRS IN NUMERATOR             C19873
C              R(3) = NUMBER OF ZERO ROOTS IN NUMERATOR                C19874
C              R(4) = NUMBER OF REAL NON-ZERO DENOMINATOR ROOTS        C19875
C              R(5) = NUMBER OF COMPLEX PAIRS IN DENOMINATOR           019876
C              R(6) = NUMBER OF ZERO ROOTS IN DENOMINATOR              C19877
C              R(7) = BODE GAIN AS DEFINED BY EQUATIONS III-46 AND 70  C19878
C                     IN VOL I. COMPUTED IN DYNSYS FROM DATA GENERATED C19879
C                     IN DFCKRD AND RETURNED IN DUMMY ARRAY 'GB'       C19880
C         FOR I=8,9,........L1    WHERE L1=7+R(1)                      C19881
C              R(I) = THE R(1) TIME CONSTANTS ASSOCIATED WITH THE REAL C19882
C                     NON-ZERO NUMERATOR ROOTS                         C19883
C         FOR I=L1+1,L1+3,......L2-1   WHERE  L2=7+R(1)+2*R(2)         C19884
C              R(I) = THE R(2) DAMPING ZETAS ASSOCIATED WITH THE       C19885
C                     NUMERATOR ROOTS WHICH APPEAR AS COMPLEX PAIRS    C19886
C         FOR I=L1+2,L1+4,......L2                                     C19887
C              R(I) = THE R(2) RESPECTIVE RESONANT FREQUENCIES         019888
C                     ASSOCIATED WITH THE NUMERATOR ROOTS WHICH APPEAR C19889
C                     AS COMPLEX PAIRS                                 C19890
C         FOR I=L2+1,......L3   WHERE L3=7+R(1)+2*R(2)+R(4)            019891
C              R(I) = THE R(4) TIME CONSTANTS ASSOCIATED WITH THE REAL C19892
C                     NON-ZERO DENOMINATOR ROOTS                       019893
C         FOR I=L3+1,L3+3,......L4-1   WHERE L4=7+R(1)+2*R(2)+R(4)+2*R(5)  019894
C              R(I) = THE R(5) DAMPING ZETAS ASSOCIATED WITH THE       C19895
C                     DENOMINATOR ROOTS WHICH APPEAR AS COMPLEX PAIRS  C19896
C         FOR I=L3+2,L3+4,......L4                                     C19897
C              R(I) = THE R(5) RESPECTIVE RESONANT FREQUENCIES         C19898
C                     ASSOCIATED WITH THE DENOMINATOR ROOTS WHICH APPEAR  019899
C                     AS COMPLEX PAIRS                                 C19900
C         (TO PRINT SET DEBUG(63)=.TRUE.. THE FOLLOWING CAPTION IS     019901
C         PRINTED)                                                     C19902
C      'SUBROUTINE TTFF ROOTS ARRAY IS'  = THE FULL ROOTS ARRAY 'R' IS C19903
C                                          PRINTED FOLLOWING THIS      C19904
C                                          CAPTION                     C19905
C                                                                      C19906
C                                                                      C19907
C         CANCER = CANCEL ROOTS COMMON TO NUMERATOR AND DENOMINATOR    C19908
C         -------------------------------------------------------      019909
C                                                                      C19910
C    R       = ROOTS ARRAY 'R' DIMENSIONED (2*L1+7) AND STORED         019911
C              IN ZEROOTZ. THE ROOTS ARRAY 'R' AS SETUP IN TTFF IS     C19912
C              EXAMINED FOR ROOTS COMMON TO BOTH NUMERATOR AND         C19913
C              DENOMINATOR. A TOLERANCE FACTOR OF 10**(-7) IS USED TO  C19914
C              COMPARE THE TIME CONSTANTS AND A STRICT EQUALITY IS USED  019915
C              TO COMPARE DAMPING ZETAS AND ASSOCIATED RESONANT        C19916
C              FREQUENCIES. WHEN COMMON ROOTS ARE FOUND THE APPROPRIATE  C19917
C              COUNTS ARE CHANGED AND THE ROOTS REMOVED FROM 'R'       019918
C         (TO PRINT SET DEBUG(64)=.TRUE. THE FOLLOWING CAPTION IS      019919
C         PRINTED)                                                     C19920
C      'ROOT ARRAY OUT OF CANCER IS' = THE FULL ROOTS ARRAY WITH ROOTS C19921
C                                      COMMON TO NUMERATOR AND         019922
C                                      DENOMINATOR CANCELLED IS PRINTED  019923
C                                      FOLLOWING THIS CAPTION          019924
C                                                                      C19925
```

```
C                                                                           019926
C         RTCP - CONSTRUCT POLYNOMIAL COEFFICIENTS ARRAY 'RX'               019927
C         ---------------------------------------------------               019928
C                                                                           019929
C     THE ALGORITHM USED FOR NUMERICAL COMPUTATION OF ROOT LOCUS DATA       019930
C         REQUIRES THAT THE TRANSFER FUNCTION BE GIVEN AS A RATIO OF        019931
C         POLYNOMIALS. THE COEFFICIENTS OF THE NUMERATOR AND DENOMINATOR    019932
C         POLYNOMIALS ARE COMPUTED IN RTCP DIRECTLY FROM THE DATA STORED    019933
C         IN THE ROOTS ARRAY 'R'                                            019934
C                                                                           019935
C     RX    = POLYNOMIAL COEFFICIENTS ARRAY 'RX', DIMENSIONED(4*L1+14)      019936
C              AND STORED IN /ZEROUT/. THE DEGREE AND THE COEFFICIENTS      019937
C              OF BOTH THE NUMERATOR AND DENOMINATOR POLYNOMIALS ARE        019938
C              STORED AS FOLLOWS (RX IS RETURNED TO DYNSAJ IN DUMMY         019939
C              ARRAY POLY)                                                  019940
C              RX(1) = THE DEGREE OF THE NUMERATOR POLYNOMIAL               019941
C              RX(2) = THE DEGREE OF THE DENOMINATOR POLYNOMIAL             019942
C         FOR I=3,.......3+RX(1)                                            019943
C              RX(I) = COEFFICIENT OF THE S**(I-3) TERM IN THE              019944
C                      NUMERATOR POLYNOMIAL                                 019945
C         FOR I=4+RX(1),5+RX(1),.....4+RX(2)+RX(1)                          019946
C              RX(I) = COEFFICIENT OF THE S**(I-(4+RX(1))) TERM IN THE      019947
C                      DENOMINATOR POLYNOMIAL                               019948
C     THE CONTENTS OF THE POLYNOMIAL COEFFICIENTS ARRAY ARE PRINTED         019949
C         UNDER THE FOLLOWING CAPTIONS ALWAYS                               019950
C     ' OUTPUT MATRIX  RDEN   (1 X ND) ' = THE ND=RX(2)+1 COEFFICIENTS      019951
C                             ASSOCIATED WITH THE DENOMINATOR POLYNOMIAL    019952
C                             FOR ASCENDING POWERS OF S                     019953
C     ' OUTPUT MATRIX  RNUM   (1 X NP) ' = THE NP=RX(1)+1 COEFFICIENTS      019954
C                             ASSOCIATED WITH THE NUMERATOR POLYNOMIAL      019955
C                             FOR ASCENDING POWERS OF S                     019956
C                                                                           019957
C                                                                           019958
C         FREQUENCY RESPONSE, EIGENVECTOR AND ROOT LOCUS COMPUTATION        019959
C         ----------------------------------------------------------        019960
C                                                                           019961
C     ALL DATA REQUIRED TO CONSTRUCT THE USER REQUESTED TRANSFER            019962
C         FUNCTION IS IN THE DATA ARRAYS R AND RX STORED IN /ZEROUT/.       019963
C         THE TRANSFER FUNCTION MAY BE EXPRESSED IN STANDARD BODE           019964
C         FACTORED POLYNOMIAL FORM FROM THE DATA IN 'R' OR IT MAY BE        019965
C         EXPRESSED IN AN UNFACTORED POLYNOMIAL FORM WITH THE DATA          019966
C         IN 'RX'.                                                          019967
C                                                                           019968
C     FOUR GRAPHICAL METHODS OF TRANSFER FUNCITON ANALYSIS ARE AVAILABLE    019969
C         FOR USER REQUEST IN DISCOS. THE DATA ASSOCIATED WITH EACH         019970
C         MAY BE EITHER PLOTTED DIRECTLY VIA USE OF THE SC4020 PLOT         019971
C         PACKAGE OR DUMPED ON TAPE FOR PROCESSING OUTSIDE OF DISCOS.       019972
C         THE GRAPHICAL REPRESENTATIONS WHICH ARE AVAILABLE FOR USER        019973
C         REQUEST ARE:                                                      019974
C               1) BODE-PLOT REPRESENTATIONS                                019975
C               2) NICHOLS CHARTS                                           019976
C               3) NYQUIST DIAGRAMS                                         019977
C               4) ROOT-LOCUS PLOTS                                         019978
C                                                                           019979
C     EIGENVECTOR ANALYSIS IN CONJUNCTION WITH FREQUENCY RESPONSE           019980
C         ANALYSIS IS NOT A COMMON PRACTISE; HOWEVER, FOR COMPLEX HIGHLY    019981
C         COUPLED MULTI-RIGID AND FLEXIBLE BODY SYSTEMS IT BECOMES A        019982
C         NECESSITY. SYSTEM ROOTS OFTEN SHIFT SIGNIFICANTLY FROM BODY       019983
C         AND CONTROL FREQUENCIES AND HENCE ARE IMPOSSIBLE TO PHYSICALLY    019984
C         INTERPERATE. EIGENVECTOR ANALYSIS OF THE CHARACTERISTIC           019985
```

```
C           MATRIX AR ASSOCIATED WITH THE TRANSFER FUNCTION UNDER STUDY      C19986
C           CAN BE USED TO DETERMINE HOW AND TO WHAT DEGREE VARIOUS BODY     C19987
C           AND CONTROL SYSTEM MODES AND FREQUENCIES BLEND TOGETHER TO       C19988
C           FORM THE TOTAL SYSTEM ROOTS AND ASSOCIATED MODES OF VIBRATION    C19989
C           (EIGENVECTORS)                                                   C19990
C                                                                            C19991
C       TO ACTIVATE THE EIGENVECTOR ANALYSIS CAPABILITY THE USER MUST        C19992
C           INPUT THE CODE WORD:                                             C19993
C                                                                            C19994
C       LEIGV = 4HEIGV  ON THE SAME DATA CARD WITH LPNAME AND LPTAPE         C19995
C                                                                            C19996
C           UPON RECOGNITION THE COMPUTER READS IN FQMIN AND FQMAX WHICH     C19997
C           DEFINE WHICH COMPLEX ROOTS OF THE CHARACTERISTIC MATRIX AR       C19998
C           REQUIRE ASSOCIATED EIGENVECTOR COMPUTATION. ALL COMPLEX ROOTS    C19999
C           HAVING IMAGINARY PART .GE. FQMIN .AND. .LE. FQMAX ARE FED INTO   C20000
C           SUBROUTINE EIGVEC WHICH COMPUTES THE ASSOCIATED EIGENVECTORS     C20001
C                                                                            C20002
C             **** PROVIDING THAT ROOTS ARE NOT MULTIPLE ROOTS ****          C20003
C                                                                            C20004
C           SOMEDAY I'LL FIND A BETTER ROUTINE FOR MULTIPLE ROOTS H.P.F      C20005
C                                                                            C20006
C                                                                            C20007
C       THE OUTPUT MATRICES PRINTED AFTER THE FOLLOWING CAPTIONS ARE:        C20008
C                                                                            C20009
C       'OUTPUT MATRIX ROOTRE IS' = REAL PARTS OF THE ROOTS OF THE           C20010
C                                   CHARACTERISTIC MATRIX AR FOR WHICH       C20011
C                                   EIGENVECTORS WILL BE COMPUTED            C20012
C       'OUTPUT MATRIX ROOTIM IS' = IMAGINARY PARTS OF THE ROOTS OF THE      C20013
C                                   CHARACTERISTIC MATRIX AR FOR WHICH       C20014
C                                   EIGENVECTORS WILL BE COMPUTED            C20015
C       'EIGENVALUE ='            = THE EIGENVALUE ASSOCIATED WITH THE       C20016
C                                   FOLLOWING EIGENVECTOR                    C20017
C       'EIGENVECTOR OF NORMAL MATRIX' = THE NZ ELEMENTS OF THE              C20018
C                                        EIGENVECTOR ASSOCIATED WITH THE     C20019
C                                        ABOVE EIGENVALUE. THE DEGREES OF    C20020
C                                        FREEDOM CONTAINED IN THE            C20021
C                                        EIGENVECTOR ARE THOSE ASSOCIATED    C20022
C                                        WITH THE Z ARRAY CONSTRUCTED        C20023
C                                        ABOVE FOR THE OPTION 'ITYPE' USED   C20024
C                                        IN SUBROUTINE ITTYPE                C20025
C       'MORE THAN 10 LOOPS FOR          = PROBABLY ROOT SENT TO EIGVEC IS   C20026
C       EIGENVECTOR OF ... DIFFERENCE      A MULTIPLE ROOT, EIGENVECTORS     C20027
C       OF ...'                            CANNOT BE FOUND. EIGENVECTOR LIST C20028
C                                        FOLLOWING PROBABLY GARBAGE          C20029
C           NOTE *** TO FIND THE DEGREE OF FREEDOM ASSOCIATED WITH EACH      C20030
C                    ELEMENT OF THE EIGENVECTOR USE 'THE TRANSFORMED         C20031
C                    STATE VECTOR CORRELATION ARRAY' DEFINED ABOVE           C20032
C                    AND THE CODE LOOPS USED FOR THE PARTICULAR ITYPE        C20033
C                    TRANSFER FUNCTION EVALUATED BY ITTYPE                   C20034
C                                                                            C20035
C       THE USER MUST INPUT VARIOUS CODE WORDS TO SPECIFY WHICH TYPE OF      C20036
C           DISPLAY IS REQUESTED AND IF THE GRAPHIC DISPLAY DATA IS TO BE    C20037
C           SCALED. THE FOLLOWING CONTROL WORDS ARE READ IN DYNS40           C20038
C       LPNAME = DISPLAY TYPE CODE WORD INTO AN ARRAY DIMENSIONED (5).       C20039
C                AFTER THE REQUESTED TRANSFER FUNCTION IS CONSTRUCTED        C20040
C                THE FIVE ELEMENT ARRAY LPNAME IS READ. THE FOLLOWING        C20041
C                CODES ARE RECOGNIZED TO SCAN:                               C20042
C           LPNAME(I)   = [illegible]   = NO DISPLAY REQUESTED               C20043
C            .........    [illegible] = [illegible] DISPLAY ONLY             C20044
C                         [illegible] NICHOLS DISPLAY ONLY                   C20045
```

```
C                       4HNYQU = GET NYQUIST DISPLAY ONLY                020046
C                       4HNINY = GET NICHOLS AND NYQUIST DISPLAYS        020047
C                       4HBONN = GET BODE, NICHOLS AND NYQUIST DISPLAYS  020048
C                       4HROOT = GET ROOT LOCUS DISPLAY                  020049
C     LPTAPE = DISPLAY DATA DUMP CODE. THIS CODE IS READ FROM THE SAME   020050
C             INPUT DATA CARD USED TO DEFINE THE LPNAME ARRAY. ALL RAW   020051
C             DISPLAY DATA IS COMPUTED IN EITHER SFREQ2 OR RLOCUS AND    020052
C             STORED IN /PSTUFF/. IMMEDIATELY UPON RETURN TO DYNS40      020053
C             FROM SFREQ2 OR RLOCUS THE FOLLOWING CODED CHECK IS MADE:   020054
C                   IF(LPTAPE.NE.4H7TRK) GO TO 1                         020055
C                   WRITE(12,100) KSAVE                                  020056
C                   WRITE(12,101) (SAVEC(I),SAVEP(I),                    020057
C                  *                        SAVED(I),SAVEA(I),I=1,KSAVE) 020058
C                 1 CONTINUE                                             020059
C               100 FORMAT (I5)                                          020060
C               101 FORMAT (4D18.8)                                      020061
C                                                                        020062
C                                                                        020063
C        FREQUENCY RESPONSE PROCESSING                                   020064
C        -----------------------------                                   020065
C                                                                        020066
C     THE FREQUENCY RESPONSE PROCESSING SECTION OF DYNS40 IS ENTERED     020067
C                 IF(LPNAME(I).EQ.4HBODE.OR.                             020068
C                    LPNAME(I).EQ.4HNICH.OR.                             020069
C                    LPNAME(I).EQ.4HNYQU.OR.                             020070
C                    LPNAME(I).EQ.4HNINY.OR.                             020071
C                    LPNAME(I).EQ.4HBONN)                                020072
C        WHERE I=1,2,.....,5. SUBROUTINE SFREQ2 DOES ALL OF THE BASIC    020073
C        COMPUTATION REQUIRED. THE USER HAS CONTROL OF THE FREQUENCY     020074
C        RANGE OVER WHICH FREQUENCY RESPONSE DATA IS TO BE COMPUTED.     020075
C        THIS IS DONE BY USER SPECIFICATION OF TWO RANGE PARAMETERS:     020076
C     FMIN   = LOWER LIMIT FOR FREQUENCY RESPONSE COMPUTATION IN SFREQ2  020077
C             A REAL*8 CONSTANT STORED IN /ZEBDATA/ DEFINED FOR EACH     020078
C             LPNAME(I),I=1,5 SATISFYING THE ABOVE CRITERION             020079
C     FMAX   = UPPER LIMIT FOR FREQUENCY RESPONSE COMPUTATION IN SFREQ2  020080
C             A REAL*8 CONSTANT STORED IN /ZEBDATA/ DEFINED FOR EACH     020081
C             LPNAME(I),I=1,5 SATISFYING THE ABOVE CRITERION             020082
C                                                                        020083
C        IF EITHER BODE PLOTS OR NICHOLS CHARTS ARE TO BE GENERATED VIA  020084
C        THE SC4020 PLOT PACKAGE ROUTINES A RANGE FOR DB MAGNITUDE OF    020085
C        THE FREQUENCY RESPONSE MUST BE GIVEN. THIS IS DONE BY USER      020086
C        SPECIFICATION OF TWO RANGE PARAMETERS:                          020087
C     DBMIN  = LOWER LIMIT FOR DB MAGNITUDE OF FREQUENCY RESPONSE DATA   020088
C             TO BE PLOTTED. A REAL*8 CONSTANT STORED IN /ZEBDATA/. USER 020089
C             DEFINED FOR EACH LPNAME(I),I=1,5 SATISFYING THE ABOVE      020090
C             CRITERION                                                  020091
C     DBMAX  = UPPER LIMIT FOR DB MAGNITUDE OF FREQUENCY RESPONSE DATA   020092
C             TO BE PLOTTED. A REAL*8 CONSTANT STORED IN /ZEBDATA/. USER 020093
C             DEFINED FOR EACH LPNAME(I),I=1,5 SATISFYING THE ABOVE      020094
C             CRITERION                                                  020095
C                                                                        020096
C        IF NYQUIST DIAGRAMS ARE TO BE GENERATED VIA THE SC4020 PLOT     020097
C        PACKAGE ROUTINE AN AMPLITUDE RANGE FOR THE FREQUENCY RESPONSE   020098
C        MUST BE GIVEN. THIS IS DONE BY USER SPECIFICATION OF TWO        020099
C        RANGE PARAMETERS:                                               020100
C     AMIN   = LOWER LIMIT FOR AMPLITUDE OF FREQUENCY RESPONSE DATA TO   020101
C             BE PLOTTED. A REAL*8 CONSTANT STORED LOCALLY. USER         020102
C             DEFINED FOR EACH LPNAME(I),I=1,5 SATISFYING THE ABOVE      020103
C             CRITERION                                                  020104
C     AMAX   = UPPER LIMIT FOR AMPLITUDE OF FREQUENCY RESPONSE DATA TO   020105
```

```
C              [illegible] POSITIVE. A REAL*4 CONSTANT STORED LOCALLY. USER    020106
C              SPECIFIES A CHOICE OF VAR(I),I=1,5 SATISFYING THE ABOVE         020107
C              CONDITIONS.                                                    020108
C                                                                             020109
C                                                                             020110
C         FREQUENCY RESPONSE DATA                                             020111
C         --------- -------- ----                                             020112
C                                                                             020113
C       THE NYQUIST AND NICHOLS FREQUENCY RESPONSE DISPLAYS SIMPLY            020114
C         PRESENT THE SAME DATA IN DIFFERENT WAYS. ALL FREQUENCY              020115
C         RESPONSE DATA IS GENERATED IN SUBROUTINE SFREQ2 AND STORED          020116
C         IN /PSTUFF/. VARIABLE INCREMENTING TECHNIQUES ARE USE IN            020117
C         THE FREQUENCY BAND FMIN TO FMAX TO ACHIEVE ADEQUATE RESOLUTION      020118
C         AROUND SYSTEM NATURAL FREQUENCIES. THE FOLLOWING REAL*4 ARRAYS      020119
C         DIMENSIONED (LB) AND STORED IN /PSTUFF/ CONTAIN THE FOLLOWING       020120
C         DATA WHEN LOADED VIA SFREQ2                                         020121
C     NSAVE  = INTEGER CONSTANT. DEFINES TOTAL NUMBER OF FREQUENCY            020122
C              RESPONSE DATA POINTS IN EACH OF THE /PSTUFF/ ARRAYS            020123
C                                                                             020124
C         VARIABLE INCREMENTING TECHNIQUES ARE USED TO CHOOSE THE VALUES      020125
C         OF 'OMEGA' AT WHICH THE TRANSFER FUNCTION 'TF' (IN BODE FORM)       020126
C         IS TO BE EVALUATED. LET:                                            020127
C                               I = SQRT(-1.0)                                020128
C         COMPLEX VALUE OF TRANSFER FUNCITON 'TF' AT FREQUENCY 'OMEGA'        020129
C                     TF(I*OMEGA) = ALPHA + I*BETA                            020130
C         AMPLITUDE                                                           020131
C                              AR = DSQRT(ALPHA**2 + BETA**2)                 020132
C         PHASE (DEGREES)                                                     020133
C                            DPHI = DATAN2(BETA,ALPHA)*57.2958                020134
C                                   0 < DPHI < 360.0                          020135
C         AMPLITUDE IN DB.                                                    020136
C                           DBELL = 8.68588961*DLOG(AR)                       020137
C                                   8.68588961 = 20*LOG(E)                    020138
C         THEN:                                                               020139
C     SAVEO  = ALL VALUES OF 'OMEGA' FOR WHICH TRANSFER FUNCTION IS           020140
C              EVALUATED                                                      020141
C     SAVEA  = ALL RESPECTIVE VALUES OF AMPLITUDE 'AR'                        020142
C     SAVEP  = ALL RESPECTIVE VALUES OF PHASE IN DEGREES 'DPHI'               020143
C     SAVEB  = ALL RESPECTIVE VALUES OF AMPLITUDE IN DB 'DBELL'               020144
C                                                                             020145
C                                                                             020146
C         ROOT LOCUS PROCESSING                                               020147
C         ---------------------                                               020148
C                                                                             020149
C     THE ROOT LOCUS PROCESSING SECTION OF DISCOS IN DYNS40 IS ENTERED        020150
C                        IF(LPNAME(I).EQ.4HROOT)                              020151
C         WHERE I=1,2,....,5. SUBROUTINE RLOCUS DOES ALL OF THE BASIC         020152
C         COMPUTATION REQUIRED FOR ROOT LOCUS PLOT GENERATION.                020153
C         TO EFFECTIVELY APPLY THIS CAPABILITY IN DISCOS THE USER MUST        020154
C         BE COGNIZANT OF THE BASIC TECHNIQUE USED FOR ROOT LOCUS             020155
C         GENERATION:                                                         020156
C          1) RLOCUS ACCEPTS A NUMERATOR POLYNOMIAL N(S) AND A                020157
C             DENOMINATOR POLYNOMIAL D(S). IT ASSUMES THAT THE RATIO          020158
C             OF THESE NUMERICALLY DERIVED POLYNOMIALS IS EITHER AN           020159
C             OPEN OR QUASI-OPEN LOOP TRANSFER FUNCTION. DISCOS DOES NOT      020160
C             CROSS CHECK THIS. RLOCUS BLINDLY ACCEPTS ANY PAIR OF            020161
C             POLYNOMIALS N(S) AND D(S).                                      020162
C          2) LET:                                                            020163
C                                    N(S)                                     020164
C                    G(S)*H(S) = K * ----  = OPEN LOOP TRANSFER FUNCTION      020165
```

```
C                            D(S)
C           [illegible]
C
C                 [illegible]
C                 [illegible] LOOP TRANSFER FUNCTION
C                 G(S) = K * (N(S)/D(S))
C           AND
C                       N(S)
C                 1 + K*---- = 0 = CHARACTERISTIC EQUATION. THE LOCUS
C                       D(S)          OF THE ROOTS PLOTTED IN THE COMPLEX
C                                     S-PLANE AS A FUNCTION OF THE OPEN
C                                     LOOP GAIN FACTOR K IS THE DESIRED
C                                     ROOT LOCUS
C        3) FOR A HIGHER ORDER SYSTEMS THE COMPUTATION OF ALL
C           ROOT LOCI IS UNNECESSARY. CONSEQUENTLY DISCUS USES AN
C           ITERATIVE PROCEDURE WHICH REQUIRES THAT THE USER DEFINE
C           THE PARTICULAR OPEN LOOP TRANSFER FUNCTION POLE OR ZERO
C           WHICH WILL BE THE STARTING POINT FOR THE ALGORITM. ONCE
C           DEFINED, THE LOCUS FROM OPEN LOOP POLE (ZERO) TO OPEN LOOP
C           ZERO (OR INFINITY) IS EVALUATED. THE FOLLOWING PARAMETERS
C           ARE SET BY INPUT DATA CARDS:
C     NRLC   = TOTAL NUMBER OF ROOT LOCI TO BE COMPUTED FOR PARTICULAR
C              OPEN LOOP TRANSFER FUNCTION DEFINED BY N(S) AND D(S)
C     IJV    = ROOT STARTING POINT ARRAY. A (2 BY NRLC) INTEGER ARRAY
C              STORED LOCALLY. FOR J=1,2,....,NRLC
C        ISNEW = IJV(1,J) = 1 IF STARTING POINT OF THE J-TH LOCUS TO BE
C                           COMPUTED IS AN OPEN LOOP ZERO; THAT IS, A
C                           ROOT OF N(S)
C                         = 2 IF STARTING POINT OF THE J-TH LOCUS TO BE
C                           COMPUTED IS AN OPEN LOOP POLE; THAT IS, A
C                           ROOT OF D(S)
C           ALL ROOTS OF THE NUMERATOR N(S) ARE LISTED UNDER THE
C           CAPTION
C                           'ZOM'
C                   'NO   REAL PART   IMAGINARY PART'
C           ALL ROOTS OF THE DENOMINATOR D(S) ARE LISTED UNDER THE
C           CAPTION
C                           'POM'
C                   'NO   REAL PART   IMAGINARY PART'
C        IS = IJV(2,J) - THE INTEGER NUMBER, TAKEN FROM THE DATA
C                        PRINTED UNDER THE ABOVE CAPTIONS, ASSOCIATED
C                        WITH THE PARTICULAR POLE OR ZERO WHICH IS TO
C                        BE USED AS THE STARTING POINT FOR THE J-TH
C                        ROOT LOCUS GENERATED
C        4) THE ROOT LOCUS ITERATION ALGORITHM IS DISCUSSED IN VOL I
C           APPENDIX C. SIX ADDITIONAL PARAMETERS MUST BE USER
C           SUPPLIED FOR EACH ROOT LOCUS. THE FOLLOWING INPUT ARRAY
C           SETS THE ITERATION CONTROL PARAMETERS
C     W1     = ITERATION CONTROL DATA ARRAY. A (5 BY NRLC) REAL*8 ARRAY
C              STORED IN THE WORK AREA IN /ZWORK1/. FOR J=1,2,....,NRLC
C      THEINC = W1(1,J) = INITIAL SEARCH ANGLE FOR J-TH LOCUS.
C                         NOMINALLY EQUAL TO 180.000 (DEGREES)
C         SCL = W1(2,J) = SCALE FACTOR FOR J-TH LOCUS. NOMINALLY EQUAL
C                         TO 1.000. IF(W1(2,J).LE.0.0) SCALE FACTOR
C                         WILL BE COMPUTED
C        ALOC = W1(3,J) = PHASE CONTROL PARAMETER EQUALS 1.000 IF GAIN
C                         FACTOR POSITIVE; EQUALS -1.0 IF GAIN FACTOR K
C                         NEGATIVE
C        XMIN = W1(4,J) = MINIMUM ADMISSIBLE REAL ROOT VALUE
C        XMAX = W1(5,J) = MAXIMUM ADMISSIBLE REAL ROOT VALUE
```

```
C        YMAX = WI(6,J) = MAXIMUM ADMISSIBLE IMAGINARY ROOT VALUE
C         BRIEFLY; THE ALGORITHM IS STARTED AT A KNOWN POINT ON THE
C            LOCUS, NAMELY AN OPEN LOOP POLE OR ZERO. A SMALL CIRCLE
C            IN THE S-PLANE ABOUT THE STARTING POINT (DIAMETER VARIED
C            INTERNALLY) IS DEFINED AND A SEARCH MADE TO FIND WHERE
C            THE LOCUS INTERSECTS THE CIRCLE. THE INTERSECTION POINT
C            IS A POINT ON THE LOCUS. A CIRCLE IS DEFINED ABOUT THE NEW
C            POINT AN THE PROCEDURE REPEATED. SINCE
C
C                            N(S)
C                      K * ---- = -1
C                            D(S)
C
C            CIRCLE INTERSECTION IS DEFINE BY THAT VALUE OF S FOR WHICH
C            THE PHASE ANGLE OF (N(S)/D(S)) = 180 DEGREES. THE OPEN
C            LOOP GAIN FACTOR K CAN THEN BE COMPUTED DIRECTLY FROM THE
C            EQUATION K = CABS(D(S))/CABS(N(S))
C
C
C        ROOT LOCUS DATA
C        ---------------
C
C     ALL ROOT LOCUS COMPUTATION IS DONE IN SUBROUTINE RLOCUS. THE USER
C        DEFINES THE NUMBER OF ROOT LOCI OF INTEREST FOR EACH TRANSFER
C        FUNCTION AND A STARTING POINT FOR THE ALGORITHM ON EACH
C        RESPECTIVE LOCUS. SYSANO CALLS RLOCUS EXACTLY NRL TIMES. EACH
C        CALL TO RLOCUS REQUESTS THE COMPUTATION OF ANOTHER ROOT LOCUS
C        ALL DATA IS STORED IN /PSTUFF/ AND MAY BE OUTPUTTED ON TAPE
C        10(OPTION-SAVE/LOAD). THE FOLLOWING ARRAYS IN /PSTUFF/ ARE
C        OUTPUT BY RLOCUS
C     KSAVE  = TOTAL NUMBER OF POINTS ON THE PARTICULAR ROOT LOCUS
C              GENERATED
C     XS(KSAVE) = ALL REAL COORDINATES OF THE KSAVE POINTS ON THE ROOT
C              LOCUS GENERATED
C     YS(KSAVE) = ALL IMAGINARY COORDINATES OF THE RESPECTIVE KSAVE POINTS
C              ON THE ROOT LOCUS GENERATED
C        IN SYSANO AFTER EACH RETURN FROM RLOCUS, THE DATA IN /PSTUFF/
C        IS EITHER PLOTTED OR DUMPED ON TAPE UNIT 10 OR BOTH
C
C     [illegible] INPUT DATA IS ALSO PRINTED FOR THE
C        [illegible] DATA PRINTED [illegible] WITH THE FOLLOWING CAPTIONS ARE:
C     'NUM' [illegible]             = POLYNOMIAL FORM OF THE
C       'A(0)      + A(1)*S+          NUMERATOR OF THE TRANSFER
C       +A(2)*S**2 + A(3)*S**3 + ...'  FUNCTION FOR WHICH THE ROOT
C                                     LOCUS IS TO BE COMPUTED A0,A1,
C                                     A2.... ARE THE COEFFICIENTS
C     'DEN' [illegible]             = POLYNOMIAL FORM OF THE
C       'B(0)      + B(1)*S+          DENOMINATOR OF THE TRANSFER
C       +B(2)*S**2 + B(3)*S**3 + ...'  FUNCTION FOR WHICH THE ROOT
C                                     LOCUS IS TO BE COMPUTED. B0,B1,
C                                     B2.... ARE THE COEFFICIENTS
C     'STARTING POINT' [illegible]    THE REAL AND IMAGINARY PARTS OF
C               X(0) + I*Y(0) '       THE STARTING POINT ON THE ROOT
C                                     LOCUS TO BE GENERATED
C     'SCAN LIMITS'                 = SCAN LIMITS FOR THE REAL PART X
C          XMIN < X < XMAX            AND IMAGINARY PART Y FOR EACH
C          YMIN < Y < YMAX '          POINT ON THE ROOT LOCUS
C                                     COMPUTATION TERMINATED IF A
C                                     LIMIT EXCEEDED
C     ' [illegible] '               = [illegible] LOOP GAIN K POSITIVE
```

```
C       '   0 DEGREE LOCUS'            = OPEN LOOP GAIN K NEGATIVE
C       'SCALE FACTOR = '              = SCALE FACTOR USED DURING
C                                        COMPUTATION OF ROOT LOCUS
C       'GAIN      ROOTS    ERROR'     = LISTED UNDER THESE HEADINGS ARE
C                                        FOR EACH POINT FOUND ON THE
C                                        ROOT LOCUS: THE OPEN LOOP GAIN
C                                        FACTOR K, THE REAL AND IMAGINARY
C                                        PART OF THE ROOT, AND THE REAL
C                                        AND IMAGINARY PARTS OF THE ERROR
C                                        MEASURE COMPUTED INTERNALLY FOR
C                                        THE ROOT
C
C
C          LINEARIZED TIME DOMAIN ANALYSIS
C--------------------------------------
C
C       THE LINEARIZED TIME DOMAIN ANALYSIS PORTION OF THE PROGRAM IS
C          ENTERED ONLY
C                        IF(LNAM.EQ.4HTIME)
C          WHERE
C      LNAM    = INPUT CONTROL WORD DEFINED ABOVE UNDER 'FREQUENCY
C                  DOMAIN ANALYSIS'
C          THE LINEARIZED DIFFERENTIAL EQUATIONS OF MOTION WHICH ARE
C          NUMERICALLY SOLVED IN SUBROUTINE LTRESP ARE OF THE FORM:
C
C              ZDT(I,T) = R**(-1)*H*R(I,J)*Z(J,T) + B(I,J)*U(J,T)
C
C          WHERE THE ELEMENTS CONTAINED IN MATRICES
C
C                               ZDT(I,T)
C                                Z(I,T)
C                         R**(-1)*H*R(I,J)
C
C          ARE DEFINED UNDER THE TITLE 'SIMILARITY TRANSFORMATION' ABOVE,
C          AND THE QUANTITY
C            B(I,J)*U(J,T) = USER SUPPLIED EXTERNAL SYSTEM INPUT ACTING
C                            ON THE I-TH PERTURBATION VARIABLE. THE USER
C                            MUST CODE THIS QUANTITY IN SUBROUTINE LTCNTL
C
C          ON PAGE [illegible], VOL I THE RECURSIVE FORMULA USED FOR
C          NUMERICAL INTEGRATION OF THE LINEARIZED EQUATIONS IS DEFINED.
C          INITIAL CONDITIONS FOR ALL PERTURBATION VARIABLES ARE ASSUMED
C          TO BE ZERO. NOTE THAT THESE EQUATIONS INCLUDE ALL NACX SENSOR
C          AND TORQUE SIGNAL EQUATIONS AS PERTURBATION VARIABLES.
C          SUBROUTINE LPRNT IS ENTERED EVERY NUMPRT INTEGRATION CYCLES.
C          A COMPLETE SELF EXPLANATORY PRINTOUT OF ALL PERTURBATION
C          VARIABLES IS PROVIDED.
C
C
C       TAPE NUMBERS AND SOME IO CONSTANTS
C-------------------------------- --------
C       THE FOLLOWING PARAMETERS DEFINE THE TAPE NUMBERS WHICH IN THE JCL
C         ARE ALLOCATED FOR USE AS SCRATCH TAPES (SHOULD BE DISK SPACE)
C         THEY ARE INITIALIZED CONSTANTS, SET IN MAIN, STORED IN /TAPENO/
C       NTAPE1 = 1
C       NTAPE2 = 2
C       NTAPE3 = 3
C
C       THE FOLLOWING ARE THE OTHER UNITS USED IN DISCUS
C       NIT        ([illegible]) DEFINED BY DATA STATEMENT
```

```
C    [illegible] DATA SET [illegible]                                020346
                                                                       020347
C    THE FOLLOWING UNITS ARE [illegible] IN THE [illegible] VERSION   020348
C      OF [illegible] SECTION                                          020349
C      UNIT 10   [illegible] TAPE FOR [illegible] TAPE INPUT           020350
C      UNIT 11   [illegible] TAPE FOR [illegible] TAPE OUTPUT          020351
C      UNIT 12   [illegible] TAPE FOR [illegible] TAPE OUTPUT          020352
C      UNIT 14   [illegible] TAPE FOR [illegible] DATA INPUT           020353
                                                                       020354
C    THE FOLLOWING [illegible] CONSTANTS SET IN MAIN                   020355
C      AND STORED IN [illegible]                                       020356
C    [illegible]         [illegible]                                   020357
C    [illegible]         [illegible]                                   020358
C    TA[illegible]       [illegible]                                   020359
C    [illegible]         [illegible]                                   020360
                                                                       020361
                                                                       020362
C    INPUT AND OUTPUT [illegible] PARAMETERS                           020363
C    ----- --- ------ ------------ ----------                          020364
C    THE FOLLOWING PARAMETERS ARE INPUTTED IN SUBROUTINE START         020365
C      AND STORED IN ZCTITLZ AS REAL*8 ARRAYS                          020366
C    NAME      [illegible] NAME READ WITH 3A8 FORMAT, DIMENSIONED (3)  020367
C    TITLE1    FIRST TITLE TO APPEAR ON EACH DATA OUTPUT PAGE OF       020368
C              [illegible], READ WITH 12A6 FORMAT, DIMENSIONED (12)    020369
C    TITLE2    SECOND TITLE TO APPEAR ON EACH DATA OUTPUT PAGE OF      020370
C              [illegible], READ WITH 12A6 FORMAT, DIMENSIONED (12)    020371
                                                                       020372
C    THE FOLLOWING PARAMETERS ARE EITHER READ OR INITIALIZED IN        020373
C      SUBROUTINE START AND STORED IN ZCSTARTZ                         020374
C    [illegible]  [illegible] READ IN IN START AS A REAL*8 PARAMETER WITH  020375
C              FORMAT A8. IF [illegible] THE DEBUG PRINT OPTIONS       020376
C              WILL BE READ IN. IF [illegible] THE PROGRAM WILL        020377
C              BE STOPPED                                              020378
C    DATE      DATE OF [illegible] SUBROUTINE DATE AS A REAL*8         020379
C              PARAMETER WITH FORMAT (A8)                              020380
C    PAGE      OUTPUT DATA PAGE COUNTER, UPDATED UPON ENTRY INTO PAGEFD.  020381
C              [illegible] VARIABLE                                    020382
                                                                       020383
C    THE FOLLOWING PARAMETER IS A REAL*8 ARRAY INPUTTED IN DYNS40      020384
C      AND STORED IN ZCTITLCZ                                          020385
C    TITLE     TITLE TO BE PRINTED WITH ALL RELAVENT FREQUENCY RESPONSE  020386
C              DATA, READ WITH FORMAT (20A4) DIMENSIONED (20)          020387
C                                                                      020388
C                                                                      020389
C    DEBUG PRINT CONTROL                                               020390
C    ----- ----- -------                                               020391
C    DEBUG     DEBUG PRINT CONTROL TABLE, INPUTTED LOGICAL*4 ARRAY     020392
C              DIMENSIONED (120) AND STORED IN ZDEBUGZ                 020393
C              IF([illegible]) THE DEBUG TABLE IS READ IN USING        020394
C              FORMAT (80L1) IN SUBROUTINE START                       020395
C              IF(DEBUG(K).EQ..TRUE.) EXERCISE DEBUG PRINT OPTIONS IN  020396
C                                      SUBROUTINE DEBUG # K            020397
C              IF(DEBUG(K).EQ..FALSE.) SKIP ALL DEBUG PRINTING IN      020398
C                                      SUBROUTINE DEBUG # K            020399
C              TO INCLUDE DEBUG PRINT STATEMENTS IN SUBROUTINE DEBUG # M  020400
C              THE FOLLOWING STATEMENTS MUST BE INSERTED               020401
C                  DATA NO/ZO/                                         020402
C                  LOGICAL DEBUG,[illegible]                           020403
C                  COMMON/ZDEBUGZ DEBUG(120)                           020404
C                  EQUIVALENCE ([illegible],DEBUG(M))                  020405
```

```
C         [illegible]                                                020406
C              [illegible] ...                                       020407
C                                                                    020408
C                                                                    020409
C     [illegible] ING                                                020410
C     [illegible] AREAS IN                                           020411
C     [illegible] WHICH MUST                                         020412
C     [illegible] ARE                                                020413
C     [illegible]                                                    020414
C                                                                    020415
C     [illegible]                                                    020416
C     [illegible]                                                    020417
C     [illegible]                                                    020418
C     [illegible] LINEAR                                             020419
C     [illegible]                                                    020420
C     [illegible] STORING,                                           020421
C     [illegible] TRANSFORMATION                                     020422
C     [illegible]                                                    020423
C     [illegible] IN THE CONSTRUCTION                                020424
C     [illegible]                                                    020425
C     [illegible]                                                    020426
C     [illegible] SENSITIVITY ANALYSIS [illegible]                   020427
C     [illegible]                                                    020428
C     [illegible] DIMENSIONED                                        020429
C     [illegible]                                                    020430
C                                                                    020431
C                                                                    020432
C     [illegible] WITHIN BLOCKS                                      020433
C     [illegible] THAT A PHYSICALLY                                  020434
C     [illegible] CONDITION HAS BEEN ENCOUNTERED)                    020435
C                                                                    020436
C     [illegible]                                                    020437
C     ------------------------ -----------                           020438
C     [illegible] AN INPUT ERROR WAS FOUND                           020439
C     [illegible]                                                    020440
C     [illegible]                                                    020441
C                                                                    020442
C     [illegible] OF THE DETERMINANT                                 020443
C     ---------------------- ------- [illegible] ZERO A ZERO GAIN    020444
C                          CONDITION ENCOUNTERED)                    020445
C     [illegible] VARIABLES STORED IN [illegible], THEY ARE          020446
C     [illegible] AND IF ZERO GAIN                                   020447
C     [illegible]                                                    020448
C     [illegible] OTHERWISE                                          020449
C     [illegible] CONDITION ENCOUNTERED; IF SO,                      020450
C     [illegible] TO THE ANALYSIS OF THE NEXT REQUESTED              020451
C     [illegible] OTHERWISE                                          020452
C                                                                    020453
C                                                                    020454
C                                                                    020455
      RETURN                                                         020456
      END                                                            020457
      SUBROUTINE EIGVEC(IVC, A, B, W, IROW, XR, XI, VR, VI, ROOTRE,     020458
     1   ROOTIE, NC, NMAX, T2, SW1, COUNTE, ERR)                        020459
C                                                       DEBUG # 117     020460
      IMPLICIT REAL*8 (A-H,O-Z)                                         020461
C        SUBROUTINE TO FIND THE EIGENVECTORS OF A NON-SYMMETRIC MATRIX  020462
C        BY A MODIFIED WILKINSON@S INVERSE ITERATION METHOD.            020463
C        CONTROL IVC CODE IS                                            020464
C              1 FIND ONLY THE REGULAR EIGENVECTORS (A X = LAMBDA X)    020465
C              2 FIND ONLY THE TRANSPOSED EIGENVECTORS (AT V = LAMBDA V) 020466
C              3 FIND BOTH TYPES OF EIGENVECTORS.                       020467
      DIMENSION A(NMAX,NMAX), B(NMAX, NMAX), W(NMAX,4), XR(NMAX),       020468
     1XI(NMAX), VR(NMAX), VI(NMAX), IROW(NMAX,2)                        020469
      INTEGER COUNT, COUNTE, T2                                         020470
      ROOTR = ROOTRE                                                    020471
      ROOTI = ROOTIE                                                    020472
      N = NC                                                            020473
      N1 = N - 1                                                        020474
      NP1 = N + 1                                                       020475
      IVC1 = IVC - 1                                                    020476
      IVC2 = IVC1 - 1                                                   020477
      COUNT = 1                                                         020478
      CLIM = 1.0D-4                                                     020479
      IF(ROOTI) 1, 60, 1                                                020480
C     COMPLEX EIGENVALUE.                                               020481
C                                                                       020482
    1 TEMP = - ROOTR - ROOTR                                            020483
      ISW = 2                                                           020484
      TEMP2 = ROOTR**2 + ROOTI**2                                       020485
      JJ = 300                                                          020486
      DO 606 I = 1, N                                                   020487
      IF(T2) 600, 613, 600                                              020488
  600 DO 602 J = 1, N                                                   020489
      JJ = JJ + 1                                                       020490
      IF(JJ - 250) 602, 601, 601                                        020491
  601 JJ = 1                                                            020492
      READ (T2) (W(LL,1), LL = 1,250)                                   020493
  602 B(I,J) = A(I,J)*TEMP + W(JJ,1)                                    020494
      GO TO 605                                                         020495
  613 DO 604 J = 1, N                                                   020496
  604 B(I,J) = A(I,J)*TEMP + B(I,J)                                     020497
  605 B(I,I) = B(I,I) + TEMP2                                           020498
  606 A(I,I) = A(I,I) - ROOTR                                           020499
      IF(T2 .NE. 0) REWIND T2                                           020500
      GO TO 700                                                         020501
  607 IF(ICJ) 622, 608, 622                                             020502
C     MATRIX SINGULAR.                                                  020503
  622 IF(IVC2) 623, 625, 623                                            020504
  623 DO 624 LL = 1, N                                                  020505
      W(LL,3) = 0.D0                                                    020506
  624 W(LL,1) = 1.D0                                                    020507
      IF(IVC1) 625, 614, 625                                            020508
  625 DO 626 LL = 1, N                                                  020509
      W(LL,4) = 0.D0                                                    020510
  626 W(LL,2) = 1.D0                                                    020511
      GO TO 611                                                         020512
C     MATRIX NOT SINGULAR.                                              020513
  608 DO 609 LL = 1, N                                                  020514
      W(LL,2) = 1.D0                                                    020515
      W(LL,3) = 1.D0                                                    020516
      W(LL,4) = 1.D0                                                    020517
  609 W(LL,1) = 1.D0                                                    020518
  610 IF(IVC1) 613, 614, 613                                            020519
  610 DO 611 I = 1, N
      I2 = INCW(I,2)
      XI(I2) = W(I,1)*ROOTI
      DO 611 J = 1, N
  611 XI(I2) = XI(I2) + A(I,J)*W(J,2)
      IF(IVEC1) 612, 500, 612
  612 DO 613 I = 1, N
      VI(I) = W(I,3)*ROOTI
      DO 613 J = 1,N
  613 VI(I) = VI(I) + A(J,I)*W(J,4)
      GO TO 495
  615 CERR = 0.D0
      IF(IVEC2) 616, 619, 616
  616 DO 618 I = 1, N
      XR(I) = -W(I,2)
      DO 617 J = 1, N
  617 XR(I) = XR(I) + A(I,J)*XI(J)
  618 XR(I) = XR(I)/ROOTI
      IF(IVEC1) 619, 633, 619
  619 DO 621 I = 1, N
      VR(I) = -W(I,4)
      DO 620 J = 1, N
  620 VR(I) = VR(I) + A(J,I)*VI(J)
  621 VR(I) = VR(I)/ROOTI
C
C     SEARCH VECTORS FOR LARGEST ELEMENT AND NORMALIZE.
C
  627 AMAX = 0.D0
      DO 629 L = 1, N
      TEMP = VR(L)**2 + VI(L)**2
      IF(TEMP - AMAX) 629, 629, 628
  628 AMAX = TEMP
      I2 = L
  629 CONTINUE
      C1 = VR(I2)/AMAX
      C2 = -VI(I2)/AMAX
      DO 630 L = 1, N
      TEMP = VI(L)
      VI(L) = VR(L)*C2 + TEMP*C1
  630 VR(L) = VR(L)*C1 - TEMP*C2
      IF(ICONT .EQ. 1) GO TO 632
      DO 631 LL = 1, N
  631 CERR = DMAX1(CERR,DABS(VR(LL) - W(LL,3)),DABS(VI(LL) - W(LL,4)))
  632 IF(IVEC2) 633, 636, 633
  633 AMAX = 0.D0
      DO 635 L = 1, N
      TEMP = XR(L)**2 + XI(L)**2
      IF(TEMP - AMAX) 635, 635, 634
  634 AMAX = TEMP
      I2 = L
  635 CONTINUE
      C1 = XR(I2)/AMAX
      C2 = -XI(I2)/AMAX
      DO 636 L = 1, N
      TEMP = XI(L)
      XI(L) = XR(L)*C2 + TEMP*C1
  636 XR(L) = XR(L)*C1 - TEMP*C2
      IF(ICONT .EQ. 1) GO TO 640
      DO 637 LL = 1, N
  637 CERR = DMAX1(CERR,DABS(XR(LL) - W(LL,1)),DABS(XI(LL) - W(LL,2)))
```

```
C
C        TEST FOR CONVERGENCE
C
  638 IF(ICOUNT .EQ. 1) GO TO 646
      IF(CONV .LE. 1.0D-8) GO TO 639
      IF(CONV .GE. C(I9)) GO TO 646
      C(I9) = CONV
      IF(C(I) .LE. 1.0D-6) GO TO 646
  639 IF(ICOUNT .GT. 40) GO TO 66
  647 ICOUNT = ICOUNT + 1
      IF(HOLD(I)) 642, 675, 642
  642 IF(INC1) 643, 644, 643
  643 DO 641 LL = 1, N
      A(LL,1) = XR(LL)
  641 A(LL,2) = XI(LL)
      IF(INC1) 644, 610, 644
  644 DO 645 LL = 1, N
      A(LL,3) = VR(LL)
  645 A(LL,4) = VI(LL)
      GO TO 655
  646 CERR = 0.0D0
      IF(ICC) 646, 647, 648
  648 ERR = CERR
      KOUNT = ICOUNT
      IF(HOLD(I)) 667, 668, 667
  667 DO 668 I = 1, N
  668 A(I,I) = A(I,I) + ROOTH
      RETURN
  66  PRINT 101, ROOTH, ROOTL, CERR
      GO TO 646
C
C        REAL EIGENVECTORS.
C
C              NOTE THAT THIS IS THE PROPER PLACE FOR STATEMENT 60
   60 CONTINUE
      ISW=1
C              STATEMENT 60 MISSING FROM DATA SUPPLIED
      DO 651 I = 1, N
      DO 650 J = 1, N
  650 E(I,J) = A(I,J)
  651 E(I,I) = E(I,I) - ROOTH
      GO TO 700
  652 IF(ICC) 680, 685, 680
C
C        SINGULAR MATRIX.
C
  680 IF(INC2) 681, 683, 681
  681 DO 682 L = 1, N
  682 XI(L) = 0.00
      IF(INC1) 683, 514, 683
  683 DO 684 L = 1, N
  684 VI(L) = 0.00
      GO TO 511
C
C        MATRIX NOT SINGULAR.
C
  685 IF(INC2) 653, 656, 653
  653 DO 654 L = 1, N
  654 XI(L) = 1.00
      IF(INC1) 656, 500, 656
  656 DO 657 L = 1, N                                                   020640
  657 VI(L) = 1.D0                                                      020641
      GO TO 655                                                         020642
C                                                                       020643
C      NORMALIZE REAL VECTORS.                                          020644
C                                                                       020645
  655 CERR = 0.D0                                                       020646
      IF(IVC2) 658, 662, 658                                            020647
  658 C1 = 0.D0                                                         020648
      C2 = 0.D0                                                         020649
      DO 660 L = 1, N                                                   020650
      TEMP =DABS(XI(L))                                                 020651
      IF(TEMP - C1) 660, 660, 659                                       020652
  659 C1 = TEMP                                                         020653
      C2 = XI(L)                                                        020654
  660 CONTINUE                                                          020655
      DO 661 L = 1, N                                                   020656
      XI(L) = XI(L)/C2                                                  020657
      CERR = DMAX1(CERR,DABS(XI(L) - XR(L)))                            020658
  661 XR(L) = XI(L)                                                     020659
      IF(IVC1) 662, 638, 662                                            020660
  662 C1 = 0.D0                                                         020661
      C2 = 0.D0                                                         020662
      DO 664 L = 1, N                                                   020663
      TEMP =DABS(VI(L))                                                 020664
      IF(TEMP - C1) 664, 664, 663                                       020665
  663 C1 = TEMP                                                         020666
      C2 = VI(L)                                                        020667
  664 CONTINUE                                                          020668
      DO 665 LL = 1, N                                                  020669
      VI(LL) = VI(LL)/C2                                                020670
      CERR = DMAX1(CERR,DABS(VI(LL) - W(LL,1)))                         020671
      VR(LL) = VI(LL)                                                   020672
  665 W(LL,1) = VI(LL)                                                  020673
      GO TO 638                                                         020674
  668 IF(IVC2) 669, 671, 669                                            020675
  669 DO 670 L = 1, N                                                   020676
  670 XI(L) = 0.D0                                                      020677
      IF(IVC1) 671, 70, 671                                             020678
  671 DO 672 L = 1, N                                                   020679
  672 VI(L) = 0.D0                                                      020680
   70 RETURN                                                            020681
C                                                                       020682
 673   IF(IVC2) 674,502,674                                             020683
 674   DO 675 I = 1,N                                                   020684
       I2 = IROW(I,2)                                                   020685
 675   XI(I2) = XR(I)                                                   020686
       GO TO 500                                                        020687
  499 IF(IVC2) 500, 502, 500                                            020688
  500 DO 501 I = 2, N                                                   020689
      I1 = I - 1                                                        020690
      DO 501 J = 1, I1                                                  020691
  501 XI(I) = XI(I) - U(I,J)*XI(J)                                      020692
  511 IF(IVC1) 502, 514, 502                                            020693
  502 DO 510 I = 1, N                                                   020694
      I1 = I - 1                                                        020695
      IF(I1) 503, 505, 503                                              020696
  503 DO 504 J = 1, I1                                                  020697
  504 VI(I) = VI(I) - U(J,I)*VI(J)                                      020698
      IF(ICC) 505, 506, 505                                             020699
```

```
505 IF(B(I,I)) 506, 507, 506                                    020700
506 VI(I) = VI(I)/B(I,I)                                        020701
    GO TO 510                                                   020702
507 IF(VI(I)) 508, 509, 508                                     020703
508 VI(I) = VI(I)*1.0D+15                                       020704
    GO TO 510                                                   020705
509 VI(I) = 1.D0                                                020706
510 CONTINUE                                                    020707
    IF(IV(2)) 514, 525, 514                                     020708
514 DO 522 I = 1, N                                             020709
    IR = NP1 - I                                                020710
    IF(I - 1) 515, 517, 515                                     020711
515 I2 = IR + 1                                                 020712
    DO 516 J = I2, N                                            020713
516 XI(IR) = XI(IR) - B(IR,J)*XI(J)                             020714
    IF(ICC) 517, 518, 517                                       020715
517 IF(B(IR,IR)) 518, 519, 518                                  020716
518 XI(IR) = XI(IR)/B(IR,IR)                                    020717
    GO TO 522                                                   020718
519 IF(XI(IR)) 520, 521, 520                                    020719
520 XI(IR) = XI(IR)*1.0D+15                                     020720
    GO TO 522                                                   020721
521 XI(IR) = 1.D0                                               020722
522 CONTINUE                                                    020723
    IF(IV(1)) 525, 529, 525                                     020724
525 DO 526 I = 2, N                                             020725
    IR = NP1 - I                                                020726
    I2 = IR + 1                                                 020727
    DO 526 J = I2, N                                            020728
526 VI(IR) = VI(IR) - B(J,IR)*VI(J)                             020729
    DO 527 L = 1, N                                             020730
    I2 = IROW(L,1)                                              020731
527 VR(I2) = VI(L)                                              020732
    DO 528 L = 1, N                                             020733
528 VI(L) = VR(L)                                               020734
529 IF(RC(I)) 613, 633, 613                                     020735
C                                                               020736
C     FACTOR MATRIX.                                            020737
C                                                               020738
700 ICC = 0                                                     020739
    SW1 = 1.0D+30                                               020740
    DO 701 LL = 1, N                                            020741
701 IROW(LL,1) = LL                                             020742
    DO 703 K = 1, N1                                            020743
    AMAX = DABS(B(K,K))                                         020744
    IMAX = K                                                    020745
    K1 = K + 1                                                  020746
    DO 702 I = K1, N                                            020747
    IF(AMAX .GT. DABS(B(I,K))) GO TO 702                        020748
    AMAX = DABS(B(I,K))                                         020749
    IMAX = I                                                    020750
702 CONTINUE                                                    020751
    IF(AMAX .LT. SW1) SW1 = AMAX                                020752
    IF(AMAX .GE. 1.0D-25) GO TO 723                             020753
    B(K,K) = 0.D0                                               020754
    ICC = ICC + 1                                               020755
    GO TO 703                                                   020756
723 IF(IMAX .EQ. K) GO TO 704                                   020757
    DO 705 J = 1, N                                             020758
    AMAX = B(K,J)                                               020759
```

```
      B(K,J) = B(IMAX,J)                                                020760
  703 B(IMAX,J) = AMAX                                                  020761
      I2 = IROW(K,1)                                                    020762
      IROW(K,1) = IROW(IMAX,1)                                          020763
      IROW(IMAX,1) = I2                                                 020764
  704 DO 707 I = K1, N                                                  020765
      IF(B(I,K)) 705, 707, 705                                          020766
  705 B(I,K) = B(I,K)/B(K,K)                                            020767
      DO 706 J = K1, N                                                  020768
  706 B(I,J) = B(I,J) - B(K,J)*B(I,K)                                   020769
  707 CONTINUE                                                          020770
  708 CONTINUE                                                          020771
      AMAX =DABS(B(N,N))                                                020772
      IF(AMAX - 1.0D-25) 712, 712, 713                                  020773
  712 B(N,N) = 0.D0                                                     020774
      SW1 = 0.D0                                                        020775
      ICC = ICC + 1                                                     020776
      GO TO 709                                                         020777
  713 IF(AMAX .LT. SW1) SW1 = AMAX                                      020778
  709 IF(ICC .EQ. ISW) GO TO 710                                        020779
      PRINT 102, ICC                                                    020780
      COUNTE = 0                                                        020781
      RETURN                                                            020782
  710 DO 711 LL = 1, N                                                  020783
      I2 = IROW (LL,1)                                                  020784
  711 IROW(I2,2) = LL                                                   020785
      IF (FDCTI .EQ. 0.D0) GO TO 652                                    020786
      GO TO 607                                                         020787
  101 FORMAT(38HOMORE THAN 10 LOOPS FOR EIGENVECTOR OF ,D12.4,          020788
     221H DIFFERENCE OF ,D12.4)                                         020789
  102 FORMAT(10H0****WARNING**** , I4, 71H ZEROS ON DIAGONAL OF FACTORED 020790
     1 MATRIX. CHECK FOR MULTIPLE EIGENVALUES. )                        020791
      END                                                               020792
```

```
      SUBROUTINE ASQR(T2, A, B, NN, NMAX)                               020793
C                                                       DEBUG * 118     020794
      IMPLICIT REAL*8 (A-H,O-Z)                                         020795
C                                                                       020796
C     PROGRAM TO SQUARE THE A MATRIX. THE MATRIX MUST BE ORIGINALLY     020797
C     IN A. IF T2 IS NONZERO, A SQUARED WILL BE STORED ON T2, AND B     020798
C     WILL BE A BUFFER WHICH MUST HAVE AT LEAST 250 WORDS. IF T2 IS     020799
C     ZERO, A SQUARED WILL BE IN B WHICH MUST BE NMAX BY NMAX.          020800
C                                                                       020801
      DIMENSION A(NMAX, NMAX), B(NMAX, NMAX)                            020802
      INTEGER T2                                                        020803
      N = NN                                                            020804
   11 IF(T2) 12, 13, 12                                                 020805
   12 REWIND T2                                                         020806
      JJ = 0                                                            020807
      DO 3 I=1,N                                                        020808
      DO 3 J=1,N                                                        020809
      JJ = JJ + 1                                                       020810
      B(JJ,1) = 0.D0                                                    020811
      DO 1 K=1,N                                                        020812
    1 B(JJ,1) = B(JJ,1) + A(I,K)*A(K,J)                                 020813
```

```
      [illegible]                                                    C20814
      [illegible]                                                    C20815
      [illegible]                                                    C20816
      CONTINUE                                                       C20817
      [illegible]                                                    C20818
      [illegible]                                                    C20819
      RETURN                                                         C20820
   [illegible] DO 14 I = 1,N                                         C20821
      DO 14 J = 1,N                                                  C20822
      F(I,J) = 0.0                                                   C20823
      DO 14 K = 1,N                                                  C20824
   14 F(I,J) = F(I,J) + A(I,K)*A(K,J)                                C20825
      RETURN                                                         C20826
      END                                                            C20827
```

B-361

```
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS77347.T055525.RV000.NFHPF015.LOOMOD       PASSED          DDNAME=PGM=*.DD                 0 EXCPS
IEF285I   VOL SER NOS= M2SCR5.
IEF285I   SYS77347.T055525.RV000.NFHPF015.S0002403     SYSIN           DDNAME=FT05F001                 0 EXCPS
IEF285I   VOL SER NOS= M2SCR6.
IEF285I   SYS77347.T055525.RV000.NFHPF015.S0002403     DELETED         DDNAME=FT05F001                 0 EXCPS
IEF285I   VOL SER NOS= M2SCR6.
IEF285I   SYS77347.T055525.SV000.NFHPF015.R0002399     SYSOUT          DDNAME=FT06F001               321 EXCPS
IEF285I   VOL SER NOS= M2SCR2.
IEF285I   SYS77347.T055525.SV000.NFHPF015.R0002400     DELETED         DDNAME=FT07F001                 0 EXCPS
IEF285I   VOL SER NOS= M2SCR4.
IEF285I   SYS77347.T055525.SV000.NFHPF015.R0002401     DELETED         DDNAME=SYSPRINT                 0 EXCPS
IEF285I   VOL SER NOS= M2SCR8.
IEF285I   SYS77347.T055525.SV000.NFHPF015.R0002402     DELETED         DDNAME=SYSUDUMP                 0 EXCPS
IEF285I   VOL SER NOS= M2SCR2.
IEF285I   M2.ZNERS.SOURCE                              KEPT            DDNAME=FT01F001               523 EXCPS
IEF285I   VOL SER NOS= T06181.
IEF373I STEP /GO      / START 77347.2045
IEF374I STEP /GO      / STOP  77347.2055 CPU   1MIN 22.83SEC MAIN  50K LCS   0K
- STEP 03 -      RETURN CODE = 0000                               STEP TIME =    1.71 MINS=(CPU=    1.38,IO=      .33)
                                            IO IN SECS. DISK=    14.80,DRUM=     .00,TAPE=    5.39,CELL=     .00,OTHR=     .24
- SURCHARGES=(DRIVES ALOC=000,TAPE MOUNTS=000,CORE=000,PAPER=017,PRIORITY=00000)SECS.   TOTAL STP TIME=    1.99 MINS.  -
IEF285I   SYS77347.T055525.RV000.NFHPF015.LOOMOD       DELETED
IEF285I   VOL SER NOS= M2SCR5.
IEF375I  JOB /NFHPF015/ START 77347.2044
IEF376I  JOB /NFHPF015/ STOP  77347.2055 CPU   1MIN 23.49SEC
-  SYSTEM=REL21.8E (11-01-77) M2                                             TIME=20.55.32.60 DATE=12-13-77
- JOB 1147-                                                      TOTAL TIME =    1.99 MINS=(CPU=    1.39,IO=      .60)
                                            IO IN SECS. DISK=    30.80,DRUM=     .00,TAPE=    5.39,CELL=     .00,OTHR=     .54
- SURCHARGES=(DRIVES ALOC=000,TAPE MOUNTS=000,CORE=000,PAPER=017,PRIORITY=00000)SECS.   TOTAL JOB TIME=    2.27 MINS.  -
 THERE WERE 01 TAPES MOUNTED FOR THIS JOB. TAPE MOUNT CHARGE WAS 30.0 MINUTES.
```

B-362

1. Report No.
NASA TP-1219

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A Digital Computer Program for the Dynamic Interaction
Simulation of Controls and Structure (DISCOS), Volume II

5. Report Date
May 1978

6. Performing Organization Code
712

7. Author(s) Carl S. Bodley, A. Darrell Devers, A. Colton
Park, and Harold P. Frisch

8. Performing Organization Report No.
G7702-F26

9. Performing Organization Name and Address

Goddard Space Flight Center
Greenbelt, Maryland 20771

10. Work Unit No.
506-17-31

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered

Technical Paper

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This document presents a theoretical development and associated digital computer program system for the dynamic simulation and stability analysis of passive and actively controlled spacecraft. The dynamic system (spacecraft) is modeled as an assembly of rigid and/or flexible bodies not necessarily in a topological tree configuration. The computer program system may be used to investigate total system dynamic characteristics, including interaction effects between rigid and/or flexible bodies, control systems, and a wide range of environmental loadings. In addition, the program system may be used for designing attitude-control systems and for evaluating total dynamic system performance, including time-domain response and frequency-domain stability analyses.

17. Key Words (Selected by Author(s))
Spacecraft attitude dynamics, Stabilization
and control, Flexible body dynamics,
Applied mathematics

18. Distribution Statement

STAR Category 18
Unclassified–Unlimited

19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price***
Unclassified	Unclassified	464	$14.50

*For sale by the National Technical Information Service, Springfield, Virginia 22161.

NASA-Langley, 1978